

# “Monitorización de movimientos de paciente durante terapia de rehabilitación”

---

Autor: Jorge Simal López

Tutor: Edwin Daniel Oña Simbaña

Leganés, 26 septiembre de 2017



Esta obra se encuentra sujeta a la licencia Creative Commons  
**Reconocimiento – No Comercial – Sin Obra Derivada**

*“If something is important enough, even if the odds are  
against you, you should still do it”*

Elon Musk.

# Agradecimientos

Este trabajo refleja muchas de las habilidades ingenieriles adquiridas, y el esfuerzo de adaptación a entornos desconocidos.

Quiero agradecer en general a mi familia y amigos el estar siempre ahí para mí aunque a veces descuidara mi afecto hacia ellos, no sé qué habría hecho sin vosotros.

Pero en especial, quiero dar las gracias por su enorme esfuerzo a mis tíos, Jesús y Mari Luz, que me han acogido como un hijo más durante este período de estudios que termino. Nunca os lo podré recompensar.

Papá, Mamá. Gracias por tantos esfuerzos y tantas oportunidades.

Hoy, un día de junio de 2017, retrocedo en el tiempo y vuelvo al primer día de universidad. Me analizo, me recuerdo, y no soy yo, era otra persona. Y me siento sublime, porque he aprendido mucho y de muchos.

Así que por todas esas experiencias, buenas y malas, que ahora componen mis fortalezas y mi sentido de la vida, a todos los que os cruzasteis y os cruzaréis en mi camino, os lo digo,

Gracias.

# Resumen

En las páginas siguientes se expone la investigación realizada y la solución propuesta para la monitorización de un paciente en rehabilitación con la intención de evitar la oclusión y mejorar la precisión, usando más de un dispositivo de captura.

Para tal fin, se combinan la librería libfreenect2, y los programas NiTE® y Matlab® para conectar dos cámaras Kinect® versión 2 y después medir los ángulos del tren superior del cuerpo humano. Todo ello integrado en una sencilla y completa interfaz de usuario.

Se analiza entonces el potencial de la solución. Los resultados muestran que esta solución es factible y resuelve el problema propuesto con ciertas tolerancias que se han medido y puesto a prueba a base de experimentos.

**Palabras clave:** libfreenect2, OpenNI, NiTE, Kinect, v2, múltiple, rehabilitación, monitorización, MATLAB, interfaz.

# Abstract

Next pages it's exposed the work path and the suggested solution to monitor a rehabilitation patient with the goal of avoiding the occlusion and improving the accuracy, using to that more than one capturing device.

Therefore, the *libfreenect2* library, NiTE<sup>®</sup> software and Matlab<sup>®</sup> are combined in order to connect two Kinect<sup>®</sup> cameras version 2 so we'll be able to measure the angles from the upper part of the body. Everything built into a simple and solid user interface.

Then, the solution power is analyzed. The results show this solution is feasible and solves the proposed issue with some margin of error measured and tested in several experiments.

**Key words:** libfreenect2, OpenNI, NiTE, Kinect, V2, multiple, rehabilitation, monitoring, MATLAB, interface.



# Índice general

|   |           |
|---|-----------|
| <b>1. INTRODUCCIÓN</b>                          | <b>11</b> |
| 1.1. Motivación                                 | 11        |
| 1.2. Objetivos del proyecto                     | 14        |
| 1.3. Estado del arte                            | 15        |
| 1.4. Estructura del proyecto                    | 19        |
| <b>2. HERRAMIENTAS UTILIZADAS</b>               | <b>20</b> |
| 2.1. Windows                                    | 20        |
| 2.2. Kinect V2                                  | 20        |
| 2.3. Microsoft Visual Studio 15 Community       | 22        |
| 2.4. libfreenect2 (OpenKinect Project)          | 22        |
| 2.5. OpenNI2                                    | 24        |
| 2.6. NITE2                                      | 25        |
| 2.7. MATLAB                                     | 28        |
| <b>3. DESARROLLO DE LA SOLUCIÓN</b>             | <b>29</b> |
| 3.1. Proceso de investigación                   | 29        |
| 3.2. Captura de datos                           | 31        |
| 3.3. Experimentos de configuración y fiabilidad | 32        |
| 3.4. Procesamiento e interfaz                   | 44        |
| 3.5. Diagramas de flujo                         | 52        |
| 3.6. Diagrama de Gantt                          | 53        |

|  |           |
|--|-----------|
| <b>4. RESULTADOS</b>                     | <b>54</b> |
| 4.1.    Ángulos realidad vs. Solución    | 54        |
| 4.2.    Múltiple cámara vs. Única cámara | 56        |
| 4.3.    Caso práctico: ejercicios reales | 58        |
| <b>5. MARCO REGULADOR</b>                | <b>60</b> |
| 5.1.    Legislación aplicable            | 60        |
| 5.2.    Estándares técnicos              | 60        |
| 5.3.    Propiedad intelectual            | 61        |
| <b>6. ENTORNO SOCIO-ECONÓMICO</b>        | <b>62</b> |
| 6.1.    Impacto socio-económico          | 62        |
| 6.2.    Presupuesto del proyecto         | 63        |
| <b>7. CONCLUSIONES</b>                   | <b>65</b> |
| 7.1.    Discusión final                  | 65        |
| 7.2.    Desarrollos futuros              | 66        |
| <b>BIBLIOGRAFÍA</b>                      | <b>67</b> |



# Índice de figuras

|   |    |
|---|----|
| FIGURA 1. EVOLUCIÓN POBLACIÓN ENTRE 1991 Y 2001 .....   | 12 |
| FIGURA 2. PREVISIÓN DE EVOLUCIÓN 2009-204 .....   | 12 |
| FIGURA 3. MARCADORES VICON .....  | 15 |
| FIGURA 4. ESQUEMA DE SISTEMA VICON .....  | 16 |
| FIGURA 5. VISIÓN DE PROFUNDIDAD DE KINECT® VERSIÓN 2 .....  | 17 |
| FIGURA 6. WINDOWS 10, SISTEMA OPERATIVO UTILIZADO .....   | 20 |
| FIGURA 7. CÁMARA KINECT® 2 Y SISTEMA DE REFERENCIA.....   | 21 |
| FIGURA 8. ADAPTADOR PARA PC .....   | 21 |
| FIGURA 9. MAPA DE ARTICULACIONES DETECTADAS POR KINECT® V2 CON EL SDK OFICIAL.....                                      | 22 |
| FIGURA 10. LOGO DE LA COMUNIDAD ABIERTA OPENKINECT .....  | 23 |
| FIGURA 11. LOGOTIPO DE OPENNI.....  | 24 |
| FIGURA 12. LOGOTIPO DE LA EMPRESA PRIMESENSE .....  | 25 |
| FIGURA 13. FORMA DEL ESQUELETO CAPTURADO POR NITE (*).....  | 26 |
| FIGURA 14. PSI POSE .....   | 27 |
| FIGURA 15. LOGO DE MATLAB .....   | 28 |
| FIGURA 16. MOVIMIENTO PERPENDICULAR AL PLANO DE LA IMAGEN .....   | 29 |
| FIGURA 17. PÉRDIDA DEL BRAZO TRAS MOVIMIENTO PERPENDICULAR.....   | 30 |
| FIGURA 18. DISTRIBUCIÓN DEL ESPACIO DE LOS EXPERIMENTOS DE CONFIGURACIÓN .....  | 32 |
| FIGURA 19. MONITORIZACIÓN A GRAN DISTANCIA.....   | 33 |
| FIGURA 20. MONITORIZACIÓN A MUY CORTA DISTANCIA .....   | 33 |
| FIGURA 21. GRÁFICO DE SEGUIMIENTO PARA ÁNGULO 0° DIFERENCIANDO LAS DISTINTAS ETAPAS DE LA SECUENCIA DE MOVIMIENTOS..... | 34 |
| FIGURA 22A. GRÁFICOS DE SEGUIMIENTO PARA DISTINTOS ÁNGULOS.....   | 35 |
| FIGURA 23A. GRÁFICOS DE SEGUIMIENTO PARA LAS PAREJAS SIMÉTRICAS DE 20, 30 Y 60 GRADOS .....                             | 38 |
| FIGURA 24. SECUENCIA DE MOVIMIENTOS DEL EXPERIMENTO .....   | 41 |
| FIGURA 25. PLANOS CORPORALES HUMANOS .....  | 41 |
| FIGURA 26. INTENTO DE ADAPTACIÓN DE ESQUELETO LATERAL (90 GRADOS) .....   | 42 |
| FIGURA 27. INTENTO DE ADAPTACIÓN DE ESQUELETO TRASERO (180 GRADOS) .....  | 43 |
| FIGURA 28. ESQUELETO CORRECTAMENTE ADAPTADO EN DOS CÁMARAS A 20 GRADOS CADA UNA.....                                    | 43 |
| FIGURA 29. MAPA DE COORDENADAS DEL SISTEMA Y CONFIGURACIÓN DE LAS CÁMARAS .....   | 44 |
| FIGURA 30. MATRIZ DE TRANSFORMACIÓN .....   | 45 |
| FIGURA 31. INTERFAZ GRÁFICA DE LA UTILIDAD DE MONITORIZACIÓN.....   | 47 |
| FIGURA 32. PESTAÑA 2JOINTS ANGLES NO MOSTRADA EN FIGURA 30 .....  | 47 |
| FIGURA 33. EJEMPLO DEL FICHERO CSV DONDE SE GUARDAN LOS ÁNGULOS DE LA APLICACIÓN .....                                  | 50 |
| FIGURA 34. PRINCIPALES PROBLEMAS CON LA CABEZA DEL ESQUELETO DE NITE® .....   | 51 |
| FIGURA 35. MAPA DE LA SOLUCIÓN .....  | 52 |
| FIGURA 36. MAPA DEL PROGRAMA USERVIEWER.....  | 52 |
| FIGURA 37. DIAGRAMA DE GANTT DEL PROYECTO (CREADO CON LA VERSIÓN GRATUITA DE TEAMGANTT®).....                           | 53 |
| FIGURA 38. EXPERIMENTO DE PRECISIÓN EN LA MEDIDA DE LOS ÁNGULOS.....  | 54 |
| FIGURA 39. GRÁFICO COMPARATIVO ENTRE ÁNGULOS REALES Y MEDIDOS CON ERROR.....  | 55 |
| FIGURA 40. PRUEBA FINAL DE UNO DE LOS EJERCICIOS REALES .....   | 59 |

# Índice de tablas

|  |    |
|--|----|
| TABLA 1. ARTICULACIONES DETECTADAS POR NITE (*) .....                                | 26 |
| TABLA 2. RESULTADOS EXPERIMENTALES EN LA MEDICIÓN DE ÁNGULOS .....                   | 55 |
| TABLA 3. RESULTADOS MEDIOS EN LA MEDICIÓN DE ÁNGULOS.....                            | 55 |
| TABLA 4. MOSAICO QUE MUESTRA LAS MEJORAS CON CÁMARA MÚLTIPLE VERSUS ÚNICA CÁMARA.... | 57 |
| TABLA 5. RESUMEN DEL TIEMPO DE TRABAJO DEL INGENIERO .....                           | 63 |
| TABLA 6. RESUMEN COSTES DE DESARROLLO .....  | 64 |

# Capítulo 1

## Introducción

### 1.1. Motivación

Hoy en día, en el mundo hay aproximadamente mil millones de personas con discapacidad. La mayoría de ellas carece de acceso a una atención médica y a servicios de rehabilitación apropiados, sobre todo en países de ingresos bajos y medianos [1].

Esto se traduce en un aumento de las dificultades de estas personas para alcanzar y mantener un buen nivel de autonomía y salud. La carencia de servicios obstaculiza su plena inclusión y participación en todos los aspectos de la vida.

La convención sobre los derechos de las personas con discapacidad, en el seno de las Naciones Unidas, establece que los países deben garantizar que las personas con discapacidad tengan acceso a servicios de salud apropiados, incluida la atención sanitaria general y servicios de habilitación y rehabilitación, y no sufran discriminación en la prestación de servicios de salud (artículo 26).

Y organizaciones como la Organización Mundial de la Salud (OMS), colaboran en la mejora de la atención médica y los servicios de rehabilitación ayudando a los países a integrarlos en la atención primaria de salud general, entre otras medidas.

Además, en los años futuros, la discapacidad será un motivo de preocupación aún mayor, pues su prevalencia está aumentando. Ello se debe a que la población está envejeciendo y el riesgo de discapacidad es superior entre los adultos mayores [2].

Por ejemplo, en España se produjo una notable reducción de población joven entre 1991 y 2001 (ver Figura 1).

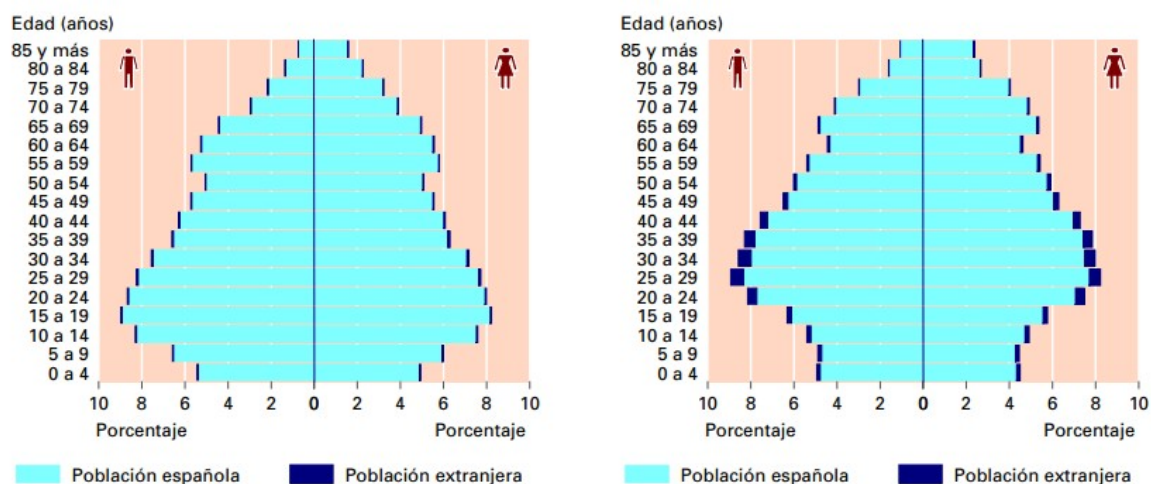


Figura 1. Evolución población entre 1991 y 2001

Y las expectativas de evolución hasta el año 2049 (ver Figura 2) nos muestran que para dicho año una gran parte de la población estará entre los 60 y 80 años.

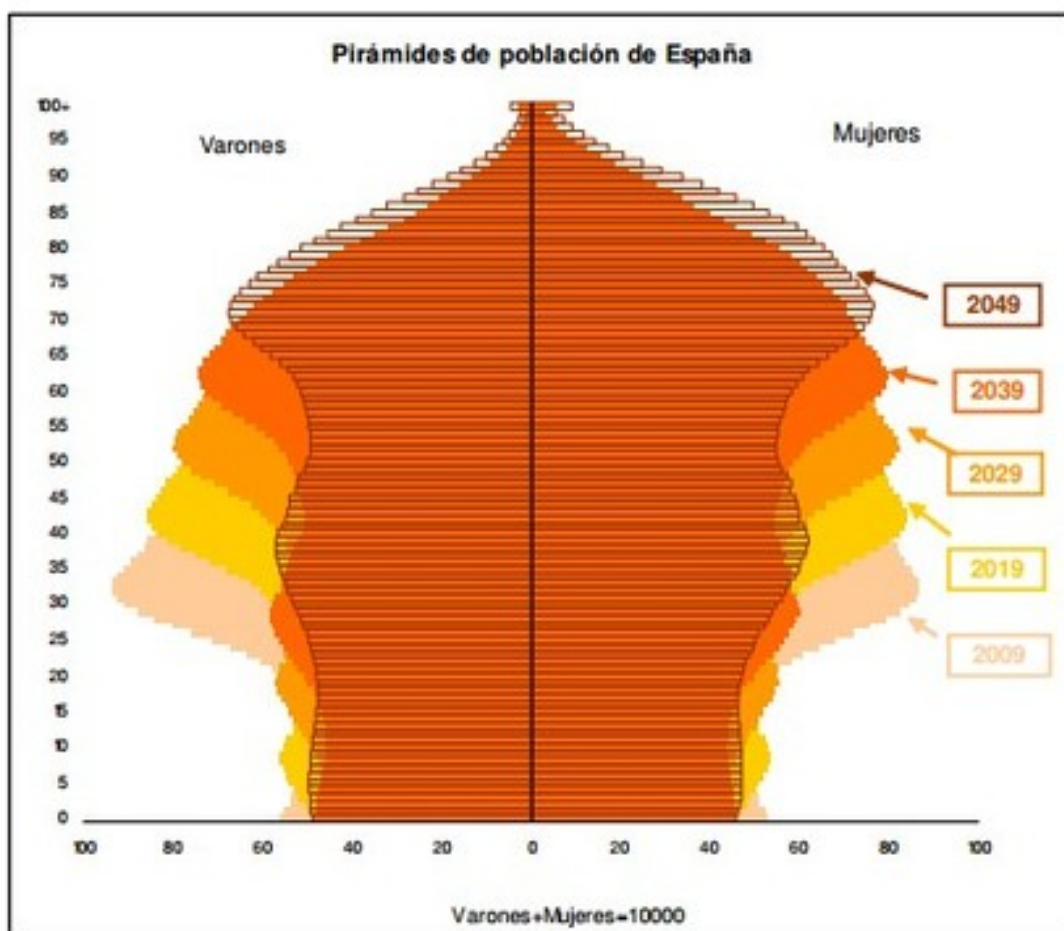


Figura 2. Previsión de evolución 2009-204

Por lo tanto, se hace indispensable una mayor investigación en este campo, el de la rehabilitación, con el fin de aumentar y mejorar las tecnologías disponibles.

Para dichos fines, hay que tener claro que una de las principales tareas en los procesos de rehabilitación es la de monitorización del proceso, no sólo para su beneficio en el momento de la terapia sino también para la valoración del progreso del paciente con el tiempo, a posteriori.

Luego sería interesante contar con un sistema de cámaras para la detección y captura del cuerpo humano que nos permita seguir sus movimientos, analizarlos y guardarlos.

Hoy en día, se utilizan sistemas Vicon® de aproximadamente 100.000€, sistemas precisos pero caros que como lógicamente no son fácil de financiar. Una gran idea o motivación sería sustituir estos caros sistemas por propuestas más asequibles.

Como ya veremos en el apartado “Estado del arte”, son ya numerosas las empresas, proyectos e investigaciones que se han desarrollado en este campo de aplicación para aprovechar el potencial de las cámaras de profundidad, o RGB-D ,como la Kinect® de Microsoft®. Intentando así lograr buena precisión a bajo coste.

Asimismo, la realidad virtual y la visión artificial son un campo en auge hoy en día, luego estas tecnologías aplicables a otros muchos campos podrían resultar de mucha más utilidad si conseguimos manejarlas con soltura.

## **1.2. Objetivos del proyecto**

El objetivo principal del presente proyecto es:

- Desarrollo de una utilidad de bajo coste para la monitorización de movimientos del tren superior de un paciente en rehabilitación asistida por un brazo robótico. El sistema intentará salvar las oclusiones e intervención del brazo robótico por medio del uso de más de un dispositivo de seguimiento sin marcadores.

Como objetivos secundarios se definen:

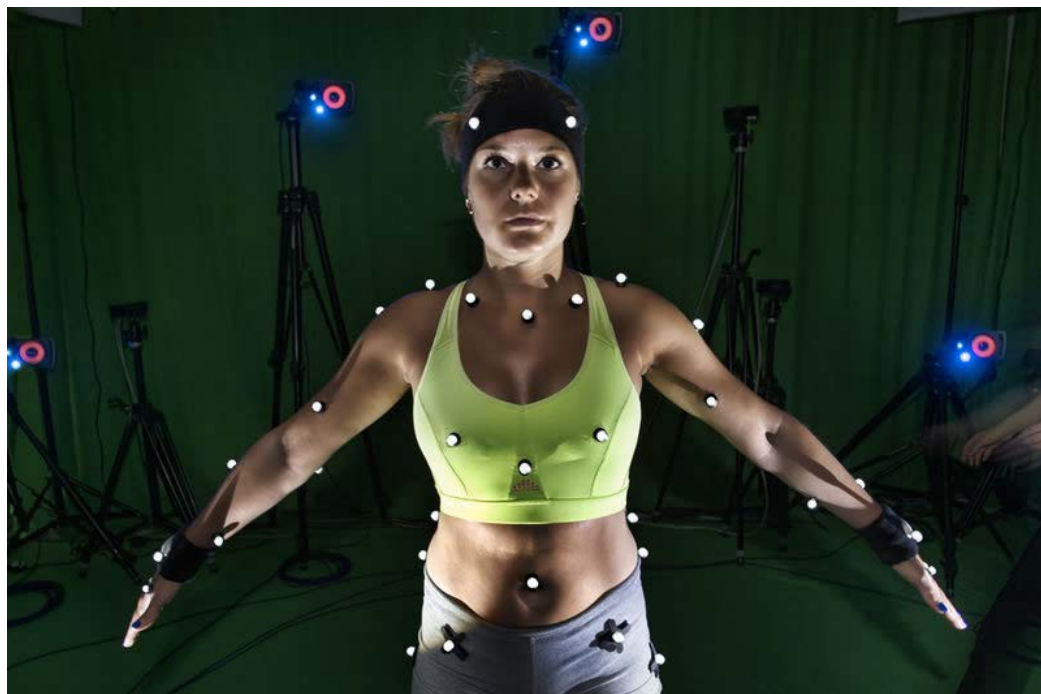
- Definir el número más adecuado de sensores para esta aplicación.
- Definir la configuración del sistema (cámaras, robot, usuario) que obtenga los resultados con mayor fiabilidad.
- Cálculo de ángulos del tren superior para soporte en ejercicios con brazo robótico del paciente en rehabilitación. Además de su almacenamiento automático para una evaluación offline.
- Búsqueda de una elevada precisión.

### 1.3. Estado del arte

La principal referencia en la tecnología de monitorización del cuerpo humano es *Vicon Motion Systems* que con más de 30 años de experiencia en el sector, es la herramienta más utilizada para grandes producciones de animación o procesos médicos, donde es necesaria la monitorización de alta precisión.

Su tecnología se basa en sistemas optoelectrónicos que registran, por medio de cámaras de luz infrarroja, los marcadores reflejantes situados en el cuerpo humano. Estos son ubicados de forma flexible dependiendo de qué se quiera monitorizar.

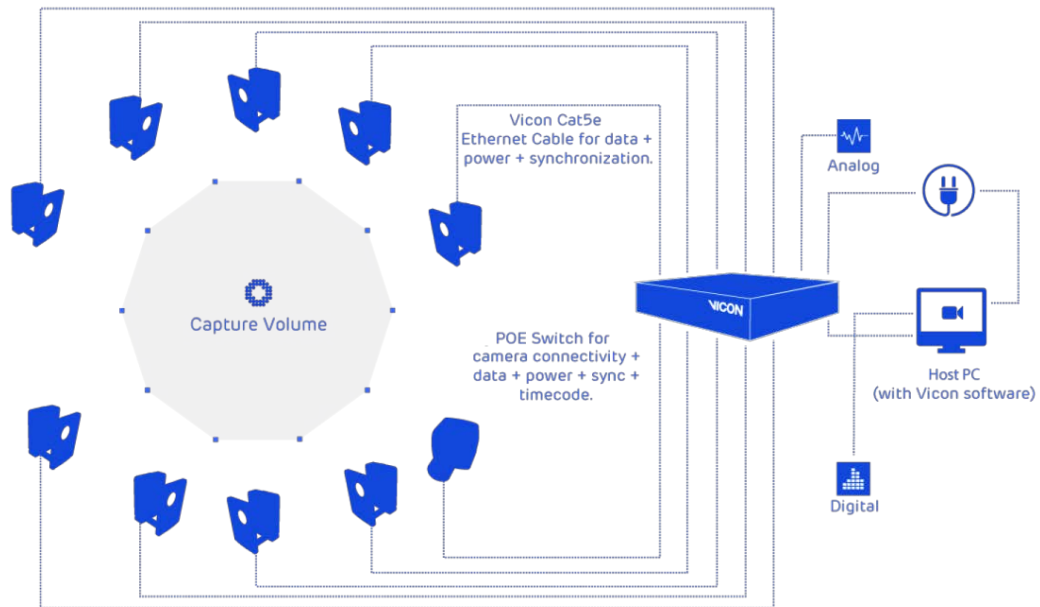
El movimiento tridimensional de cada marcador es reconstruido a partir de sus coordenadas bidimensionales en cada fotograma, coordenadas recogidas por cada cámara simultáneamente. Después, con un modelo biomecánico definido con anterioridad, se es capaz de reconstruir dicho movimiento en 3D. La sensibilidad a la colocación y calibración de los marcadores es especial importante [3], por lo que existe una fase previa a los ejercicios de generalmente una larga duración.



**Figura 3.** Marcadores Vicon

En la figura anterior (Figura 3) se pueden ver los marcadores reflectantes citados anteriormente y las cámaras traseras durante un proceso de animación. Y a

continuación (Figura 4) se presenta un esquema del despliegue de un sistema Vicon corriente, formado por 10 cámaras, un concentrador y un PC con software dedicado.



**Figura 4.** Esquema de sistema Vicon

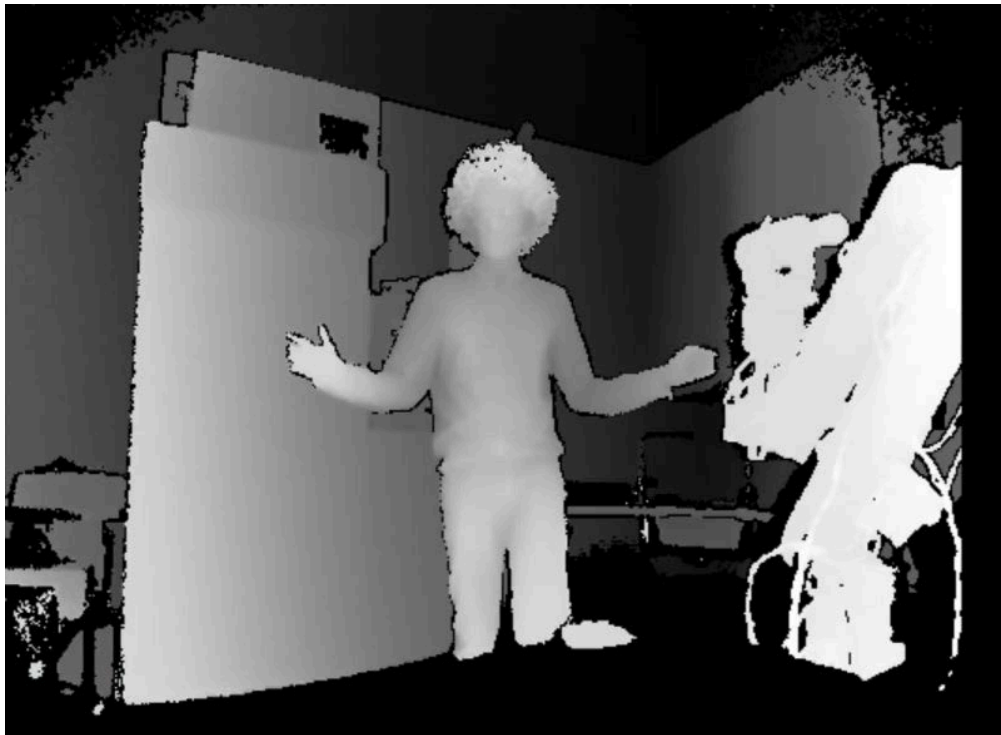
Esta tecnología destaca por su alto número de cámaras utilizadas para no perder ningún ángulo de visión y el potente software propio relacionado, así como por la fase previa de calibración y posicionamiento de los marcadores de la que ya se hablaba párrafos arriba.

Otro competidor sería Qualysis®, una empresa menos popular que Vicon® pero con la misma esencia: similares productos y tecnologías.

Ahora, céntrese la atención en las cámaras Kinect®, cámaras que con su tecnología de profundidad (ver Figura 5) bien podrían llegar a realizar funciones similares.

Entonces, ¿qué se ha hecho y qué se puede hacer con estos dispositivos en el campo de la rehabilitación? ¿Podrían competir con los sistemas Vicon®?





**Figura 5.** Visión de profundidad de Kinect® versión 2

Hay estudios muy interesantes sobre la precisión y confianza del dispositivo Kinect® V2, los cuales se van exponiendo brevemente a continuación.

Frente a un sistema Vicon®, un estudio concluye que Kinect® tiene el potencial para ser usado como herramienta clínica de medida en la que se puede confiar. Siendo además más portable e independiente de marcadores corporales. Los resultados muestran que la precisión de los movimientos fue entre moderada y excelente, con remarcable ruido en pies y rodillas [4].

Otro estudio enfocado a medir la precisión del sensor Kinect® V1 ya apuntaba lo anterior, aunque se destaca la imposibilidad de monitorizar pequeños movimientos como el cierre de las manos. Kinect® V2 lo detectaría mejor, sobre todo al incluir los pulgares como “articulación” a tener en cuenta (disponible con el software oficial) [5].

Entre estos y otros estudios consultados, se ha encontrado que uno de los principales problemas es la oclusión de algunas articulaciones en ciertos movimientos. Lo cual es solucionado muchas veces con la utilización de múltiples cámaras.

Al buscar información de esa opción de múltiples cámaras, el SDK (*Software Development Kit*) oficial de Microsoft no da la posibilidad de acceder a múltiples cámaras de versión 2 [6][7]. Entre otras razones por el gran ancho de banda necesario. Con su mayor resolución, Kinect V2 procesa 2 gigabits de datos por segundo para leer su entorno.

Al respecto, se destaca la investigación de la universidad de Hong Kong [8]. Este trabajo de investigación es bastante completo, se utiliza la SDK de Microsoft ejecutando dos procesos a la vez pero con la versión anterior de Kinect (versión 1) que sí permite conectar varias cámaras en el mismo ordenador simultáneamente.

Es remarcable aquí el procesamiento del esqueleto resultante, pues implementa un algoritmo basado en la fijación de la longitud de las extremidades para conseguir un esqueleto más preciso.

En esta línea, hay otras investigaciones como la desarrollada en la Universidad Hanyang de la República de Corea cuyo objetivo es el mismo pero utilizando otro método, la aplicación de un filtro Kalman [9].

El filtro Kalman es utilizado comúnmente en la guía, navegación y control de vehículos, especialmente naves espaciales. Es un algoritmo recursivo de predicción de sistemas dinámicos lineales [10].

Así, se puede predecir de forma aproximada dónde estará el esqueleto en el instante siguiente, pudiendo adaptar los datos obtenidos a la predicción consiguiendo corregir los fallos graves de detección que pudieran ocurrir.

De forma más sencilla, un proyecto de una universidad de Bangkok, Tailandia, realiza un análisis de postura al andar evitando la oclusión y utilizando múltiples cámaras, eso sí, de la versión 1 [11]. Esta solución es especialmente interesante por su simplicidad y buenos resultados.

Todas estas herramientas desarrolladas y alguna más servirán como inspiración para crear la solución que se requiere.

## 1.4. Estructura del proyecto

Para facilitar la lectura y aclarar la conexión entre las diferentes partes, se describe la estructura que seguirá el presente proyecto.

- En este **Capítulo 1** se presentaban las motivaciones, objetivos y el estado de la técnica.
- En el **Capítulo 2** se describirán brevemente las herramientas finalmente utilizadas tras el análisis del estado del arte, herramientas que nos permitirán desarrollar nuestro proyecto.
- El **Capítulo 3** compondrá el grueso del trabajo describiendo las fases, experimentos, y en general todo el proceso de desarrollo de soluciones y obtención de resultados.
- Los **Capítulos 4 y 5** tratan de los temas legales y del impacto socio-económico respectivamente, una visión del proyecto desde un punto de vista legal y financiero.
- Por último, el **Capítulo 6** promete ser uno de los más importantes, pues allí se analizarán las conclusiones tras el desarrollo que mostrarán los fallos, los logros y las posibilidades a futuro.

## Capítulo 2

# Herramientas utilizadas

### 2.1. Windows

No se va a entrar en detalles del sistema operativo más utilizado en el mundo porque es de sobra conocido, pero se quería hacer un inciso.

Aunque podría haberse desarrollado el proyecto en otros sistemas operativos como Ubuntu (muy utilizado por el mundo informático), se ha elegido Windows 10 (ver Figura 6) para una mayor flexibilidad a la hora de utilizar otros programas o exportar la herramienta (si es lo suficiente buena) para su uso profesional, pues prácticamente la mayoría de personas e instituciones manejan Windows cotidianamente.



**Figura 6.** Windows 10, sistema operativo utilizado

### 2.2. Kinect V2

Existen otras cámaras RGB-D y también una versión anterior de Kinect® pero se ha elegido Kinect® 2 por ser una de las más populares, gozando de un avanzado estado de investigación, y ser más preciso que su antecesor al tener mejores sensores. Esto nos podría permitir mejorar los resultados que ya se han conseguido en otras investigaciones como las que se exponían en el capítulo anterior.

Como el sensor Kinect® nació como periférico de la consola Xbox® se necesitará el adaptador para PC que fue lanzado posteriormente. Sensor y adaptador se muestran a continuación (ver Figura 7 y 8).



**Figura 7.** Cámara Kinect® 2 y sistema de referencia

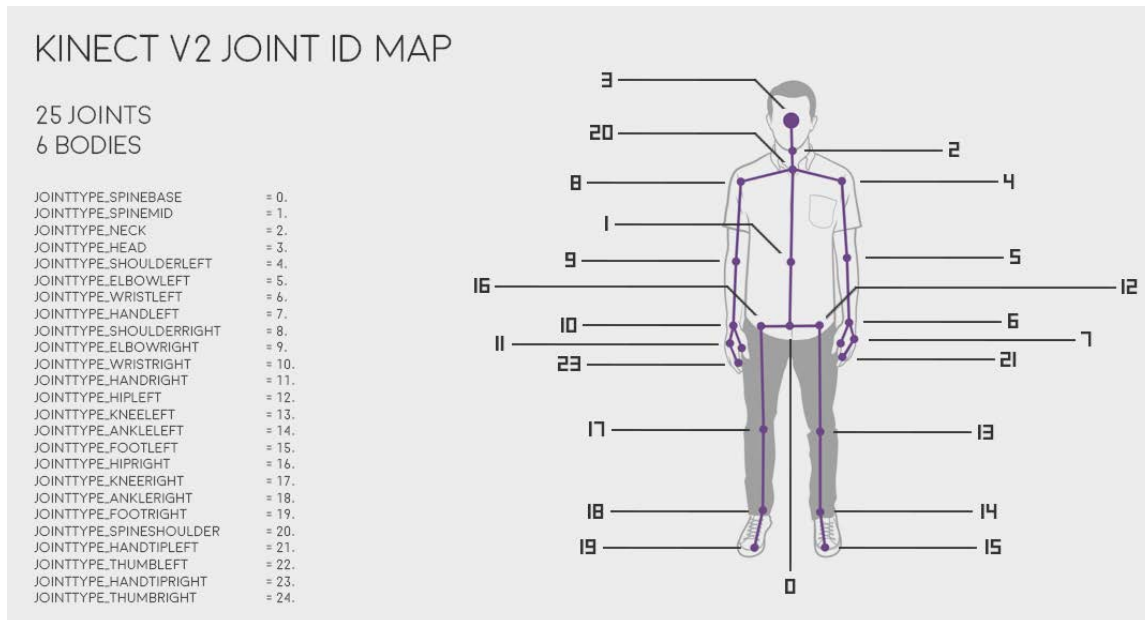


**Figura 8.** Adaptador para PC

También en la Figura 7 se observa el sistema de referencia en el que están basados los datos que podremos obtener del dispositivo, donde  $Y$  será la altura,  $Z$  la profundidad y  $X$  representará el movimiento lateral. Las medidas estarán en metros (m).

Esta cámara procesa 2 gigabits de información por segundo para leer el entorno, posee gran angular y tecnología de tiempo de vuelo. Esta tecnología consiste en proyectar una onda de luz infrarroja, la cual rebota en los objetos y vuelve con una cierta fase dependiendo de la distancia del objeto al sensor. Esta tecnología puede usarse incluso en la oscuridad, de hecho funciona mejor con baja iluminación [12].

Con el SDK oficial, se pueden detectar hasta 6 esqueletos, el ritmo cardíaco, la expresión facial, y la posición y orientación de hasta 25 articulaciones incluyendo los pulgares, tal y como muestra la Figura 9.



**Figura 9.** Mapa de articulaciones detectadas por Kinect® V2 con el SDK oficial

Además su cámara a color captura con calidad 1080p y cuenta con un micrófono que potencia sus funcionalidades.

## 2.3. Microsoft Visual Studio 15 Community

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, de sus siglas en inglés) donde se puede programar en varios lenguajes como C++, C# o .NET.

Trabajando en Windows, pudiendo adquirir una licencia gratuita y soportando el lenguaje C++ (imprescindible para desarrollar nuestra solución desde libfreenect2), se elige como el entorno de desarrollo perfecto para nuestro proyecto.

## 2.4. libfreenect2 (OpenKinect Project)

OpenKinect es una comunidad abierta de gente interesada en usar el hardware de Kinect® de Xbox® con ordenadores y otros dispositivos. Trabajan en librerías de código abierto y gratuitas para permitir que la cámara pueda ser usada en Windows, Linux y Mac [13].



Figura 10. Logo de la comunidad abierta OpenKinect

La comunidad está formada por 2000 miembros que contribuyen con su tiempo y código en el proyecto. Su principal proyecto es *libfreenect*, un programa o conjunto de librerías para Kinect® versión 1.

Por continuación, *libfreenect2* es el sucesor para la versión 2 de Kinect®. Y mientras que *libfreenect* podía acceder prácticamente a todas las características de la primera versión del sensor (audio, led, imágenes rgb y profundidad, acelerómetro y motores). *Libfreenect2* sólo puede acceder a la imagen de rgb, de IR y la de profundidad; y la “registration” entre RGB y profundidad [14].

Esto de “registration” hace referencia a la posibilidad que ofrece el sensor de rápidamente obtener la imagen RGB y de profundidad superpuestas.

Características como el audio son inaccesibles desde estos drivers y librerías.

El proceso de instalación se detalla en su página de GitHub [15], el cual, si no se está familiarizado con los procesos de desarrollo informáticos no es fácil de seguir, pues para ello hay que utilizar programas como CMake.

Además de las limitaciones de uso impuestas antes, también se necesita de un hardware potente y específico. Existen muchos usuarios y desarrolladores que han sufrido problemas con tan delicados requisitos [20].

Para la conexión con una cámara se necesita un controlador USB 3.0, pero no todos valen. Por ejemplo, los controladores Intel® y NEC® se saben que funcionan bien, y que los ASMedia® por lo contrario no funcionan con *libfreenect2*.

Para que sea posible la conexión de nuestras múltiples Kinect® al mismo ordenador a parte de los controladores, se recomiendan las siguientes especificaciones:

- Usar Windows 7 (buggy), Windows 8, Windows 8.1 o Windows 10. O también Mac OS X, Debian, Ubuntu 14.04 o mayor con kernel 3,16+ o el más actual.

- Usar una tarjeta de expansión PCI Express USB3 (con controlador NEC) para cada una de las cámaras, las cuales no pueden estar conectadas con ancho de banda distinto al de 8 u 16 slots.
- En Linux, es posible que se tenga que ampliar la memoria USBF de los buffers.

En definitiva, lo que nos permite esta utilidad es acceder al hardware; leer los puertos, crear los canales de comunicación, etc. Bajo ciertas limitaciones.

## 2.5. OpenNI2



Figura 11. Logotipo de OpenNI

OpenNI, acrónimo del inglés *Open Natural Interaction*, es una organización industrial sin ánimo de lucro y de código abierto centrada en la certificación y mejora de la interoperabilidad de interfaces naturales y orgánicas de usuario para dispositivos de interacción natural (NI), aplicaciones de estos dispositivos y el *middleware* que facilita el acceso a y uso de tales dispositivos [16].

Middleware es lo que se conoce como un programa que permite a una aplicación interactuar o comunicarse con otras aplicaciones, paquetes de programas, redes, hardware y/o sistemas operativos.

La historia de OpenNI, que nace en 2010 con la empresa PrimeSense como miembro principal, rápidamente gana adeptos y es altamente utilizado en el mundo académico y de aficionados. Pero tiene su fin en 2014 con la compra de PrimeSense por Apple Inc., que conlleva el cierre del sitio OpenNI.org.

Actualmente, está respaldado por la empresa Occipital y otros socios fundadores de PrimeSense ya que, por ejemplo, el dispositivo Structure de Occipital está creado basado en este entorno [17].



## 2.6. NiTE2

De la misma forma que OpenNI®, el algoritmo NiTE® fue creado por PrimeSense y por tanto, tras su compra por parte de Apple Inc., este desapareció junto con su página web.



Figura 12. Logotipo de la empresa PrimeSense

Sin embargo, como era software de código abierto y fue altamente utilizado y compartido por la comunidad, aún puede encontrarse la última versión en Internet.

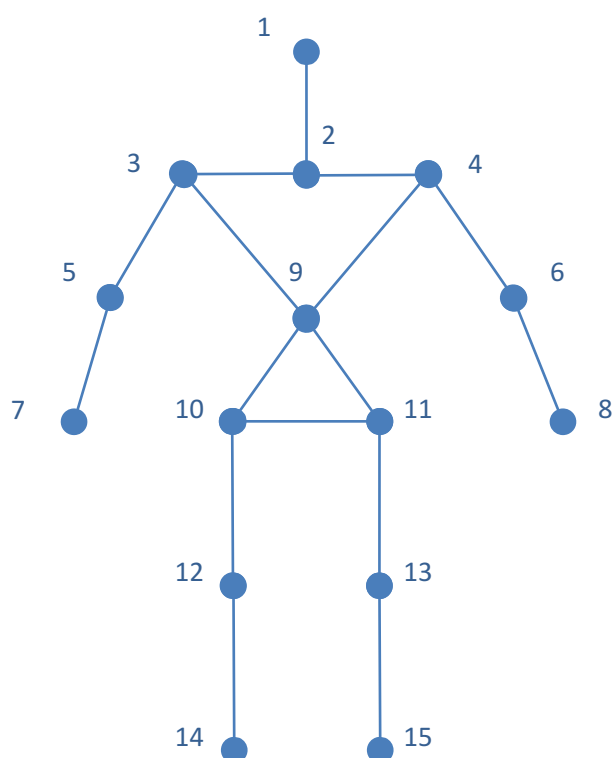
NiTE® se define o se ha definido hasta ahora como el middleware 3D más avanzado y robusto, con soporte multiplataforma y carga mínima de la CPU.

Proporciona a la aplicación un control de usuario a través de la mano o de todo el cuerpo. Nunca se ha liberado información sobre el algoritmo utilizado para tal función, pero se sabe que utiliza la profundidad, el color o el IR para detectar cuerpos y adecuarles un esqueleto.

Con todos esos “inputs”, es capaz de localizar y monitorizar manos, analizar la escena (separando a los usuarios del entorno), monitorizar hasta 15 articulaciones del esqueleto del usuario con cierta precisión, reconocer ciertos gestos, etc.

Puede encontrarse una copia de este *middleware* en un sitio de dominio ruso que parece una copia de lo que fue la página oficial de OpenNI [18].

Centrándonos en la monitorización de las articulaciones, la cual es núcleo de nuestro proyecto, se exponen a continuación en la Figura 13 y la Tabla 1 las articulaciones con su nombre y número detectadas por el algoritmo.



**Figura 13.** Forma del esqueleto capturado por NiTE (\*)

| Número | Nombre               | Descripción       |
|--------|----------------------|-------------------|
| 1      | JOINT_HEAD           | Cabeza            |
| 2      | JOINT_NECK           | Cuello            |
| 3      | JOINT_LEFT_SHOULDER  | Hombro izquierdo  |
| 4      | JOINT_RIGHT_SHOULDER | Hombro derecho    |
| 5      | JOINT_LEFT_ELBOW     | Codo izquierdo    |
| 6      | JOINT_RIGHT_ELBOW    | Codo derecho      |
| 7      | JOINT_LEFT_HAND      | Mano izquierda    |
| 8      | JOINT_RIGHT_HAND     | Mano derecha      |
| 9      | JOINT_TORSO          | Torso             |
| 10     | JOINT_LEFT_HIP       | Cadera izquierda  |
| 11     | JOINT_RIGHT_HIP      | Cadera derecha    |
| 12     | JOINT_LEFT_KNEE      | Rodilla izquierda |
| 13     | JOINT_RIGHT_KNEE     | Rodilla derecha   |
| 14     | JOINT_LEFT_FOOT      | Pie izquierdo     |
| 15     | JOINT_RIGHT_FOOT     | Pie derecho       |

**Tabla 1.** Articulaciones detectadas por NiTE (\*)

(\*) Información extraída de los códigos de NiTE®.

Bien, lo primero que ocurre es un proceso de calibración, donde el algoritmo intenta adaptar el esqueleto al cuerpo del usuario. Es un proceso de “machine learning” donde se tendrá más precisión con el tiempo hasta llegar a niveles de confianza del 100% (ver POSITION CONFIDENCE en las próximas líneas).



Figura 14. PSI Pose

NOTA: la calibración se puede poner en automático con ese proceso de “machine learning” o requerir una pose inicial fija, normalmente “PSI Pose” (ver Figura 14).

Ahora bien, el algoritmo es capaz de proporcionarnos cuatro datos fundamentales sobre cada articulación:

- **JOINT POSITION:** coordenadas cartesianas en “real world” de la articulación en el espacio. Es decir, ya se ha pasado de las coordenadas de la cámara a las del mundo real.
- **POSITION CONFIDENCE:** valor entre 0 y 1 que indica cómo de seguro está el algoritmo de la posición de la articulación. Puede tener cuatro valores:
  - “0”: la articulación queda fuera de la imagen.
  - “0,5” ó “0,7”: la articulación ha sido estimada debido a una oclusión temporal o el proceso de calibración no ha aprendido lo suficiente todavía.
  - “1”: la articulación está “trackeada”, visible y bien determinada.
- **JOINT ROTATION:** cuaternio que indica la rotación actual de la articulación, de tal forma que podríamos monitorizar el giro de las manos por ejemplo.
- **ROTATION CONFIDENCE:** indicador entre 0 y 1 que indica de la misma forma que antes la confianza de la rotación estimada por el algoritmo.

Con esta materia prima, el programa ejemplo “*UserViewer*” que viene con el algoritmo, ya grafica el esqueleto. Luego con estos datos se podría construir un esqueleto propio.

Cabe comparar la Figura 13 con la Figura 9 para ver las 25 articulaciones que ofrece el SDK oficial frente a las 15 que ofrece NiTE®. Esto puede indicar que el algoritmo de Microsoft sea más potente y fiable, pues es capaz de ir más allá y detectar incluso los pulgares.

## 2.7. MATLAB

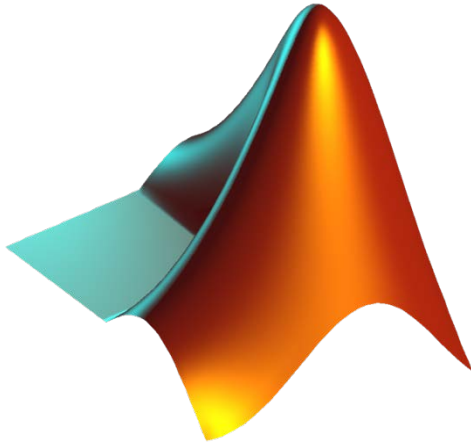


Figura 15. Logo de MATLAB

MATLAB, programa muy conocido en el mundo científico e ingenieril por ser la forma más natural del mundo para expresar las matemáticas computacionales, ya que está basado en matrices.

Luego es fácil lidiar con datos que en otros lenguajes como C++ sería más difícil (el trato de matrices en C++ es tedioso y poco flexible).

Sus gráficos integrados facilitan la visualización de los datos y la obtención de información a partir de ellos.

Además, una vasta librería de “toolboxes” preinstaladas le permiten empezar a trabajar inmediatamente con algoritmos esenciales.

El entorno de escritorio invita a experimentar, explorar y descubrir. Siendo muy “user-friendly” y dando la posibilidad de diseñar fácilmente interfaces con su herramienta GUIDE.

Para más información ver [19].

Se quiere recalcar la potencia de procesamiento que MATLAB ofrece, pues es la razón principal por la que se ha elegido este software para el proyecto. Así como por la posibilidad de creación de interfaz.

## Capítulo 3

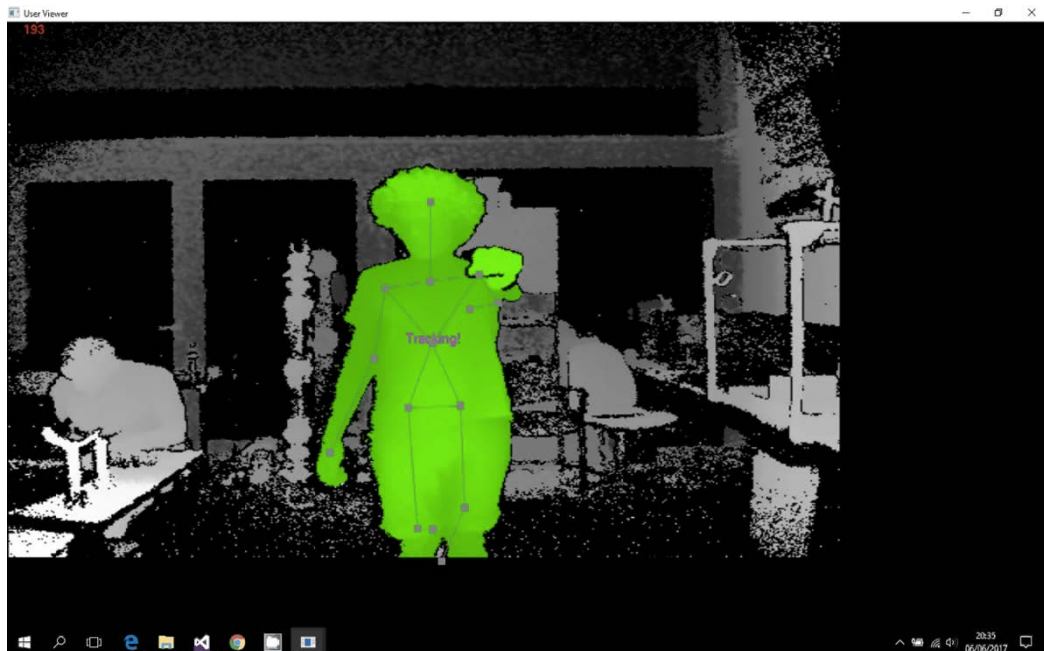
# Desarrollo de la solución

Este capítulo constituye una de las partes más importantes del proyecto, se describe en él la metodología utilizada. Dicha metodología quedará esquematizada en los diagramas de flujo del quinto apartado, el apartado 3.5, y el diagrama de Gantt del 3.6.

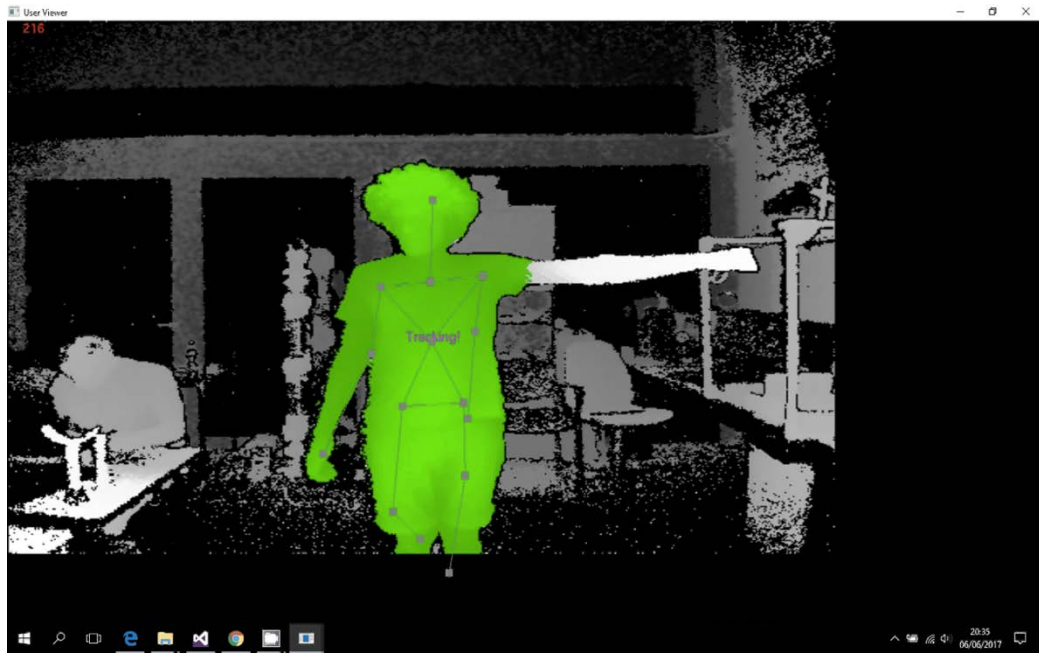
### 3.1. Proceso de investigación

Como ya veíamos en el estado del arte (Capítulo 1, 1.3), para prevenir el principal problema de la oclusión es conveniente utilizar al menos dos Kinect®.

Con el fin de ilustrar el fenómeno de la oclusión, se muestran las capturas de un ejercicio en el que se pierde el brazo tras su posicionamiento en perpendicular al plano de la cámara (ver Figura 16 y 17).



**Figura 16.** Movimiento perpendicular al plano de la imagen



**Figura 17.** Pérdida del brazo tras movimiento perpendicular

Para resolver tal problema con la utilización de varios dispositivos, se comenzó investigando las posibilidades de la utilidad oficial de Microsoft® para desarrolladores.

Dicha utilidad no ofrecía ninguna posibilidad para el acceso a múltiples dispositivos desde el mismo ordenador, el mismo equipo de Microsoft® lo deja claro en uno de sus video-tutoriales [7].

Luego tras una larga búsqueda, o se utilizan dos ordenadores interconectados con software oficial o se utiliza la librería *libfreenect2* dentro del proyecto de código abierto *OpenKinect*.

Es esta última la utilizada, por ser la que menos recursos necesita (sólo hay que comprar un ordenador) se apuesta por ella y se intenta sacarle el máximo partido.

Contábamos para ello con un ordenador personal MacBook Air de Apple que se ajusta perfectamente a los requisitos de hardware de la librería, todo era compatible.

Entonces se comenzó a explorar por los comentarios en Github sobre el tema de “Multiple Kinect”, acabando en un hilo de una bifurcación de *libfreenect2* [19]. En él se describía un programa que conecta dos cámaras y mostraba la vista RGB, la de profundidad y la infrarroja de cada una a través de OpenCV.

Lo probamos, nos funcionó, y así entendimos cómo funcionaba más o menos libfreenect2.

Después intentamos modificar el programa ejemplo que venía con libfreenect2 llamado “Protonect” con la misma filosofía de extraer visualizaciones como con OpenCV, pero esta vez no todas las ventanas se mostraban, parecía que había algún fallo con OpenGL que no permitía manejar tanta información a la vez en un mismo programa.

Ahora que teníamos una idea de cómo abrir comunicaciones con las cámaras, comenzamos a investigar la mejor forma de detectar el cuerpo humano y monitorizar sus movimientos.

Así descubrimos NiTE, enlazable con los protocolos de libfreenect2 gracias a un driver de OpenNI. Dicho driver ya se instaló en el proceso de instalación descrito en [15].

Se instaló entonces NiTE y se enlazó todo lo necesario para continuar con la lectura y comprensión en profundidad del código “NiTE.h” para ver lo que podía ofrecer. Después, se pasó a testear los ejecutables de ejemplo que vienen con NiTE®, en concreto, “*UserViewer*”.

Dicho ejemplo, hacía lo que buscábamos: detectar a la persona y su esqueleto para realizar la monitorización de sus movimientos.

### 3.2. Captura de datos

Una vez encontrado el ejemplo “*UserViewer*”, se intentó modificarlo para que abriera dos cámaras, lo que podía hacerse enviando a través de la línea de comandos de la consola que se ejecutaba el dispositivo a abrir.

Detectada esta posibilidad, se modificó el programa para crear dos ejecutables con el dispositivo a abrir especificado dentro del código de cada ejecutable.

Así, cada cámara (Kinect V2 #1 y #2) es llamada mediante su URI (Uniform Resource Identifier). Ese URI es “freenect://0” para la primera cámara y “freenect://1” para la segunda.

Si sólo una está conectada se accede a ella mediante “freenect://0”. Si las dos están conectadas, se ordenan por su número de serie, “freenect://0” será la de menor número de serie y “freenect://1” la de mayor.

Se creó entonces en el programa también una matriz que guardaba las posiciones de las articulaciones, y después una función que guardaba dicha matriz en un archivo de texto (.txt) fácilmente importable por cualquier otro programa como Excel o MATLAB para su procesamiento.

### 3.3. Experimentos de configuración y fiabilidad

Ahora que también hemos conseguido obtener las articulaciones, nos surge la problemática de encontrar la mejor configuración física de las cámaras para la captación.

Con tal fin, llevamos a cabo diferentes experimentos situando la cámara a la misma distancia pero a diferentes ángulos y realizando el mismo ejercicio con el brazo derecho. La distribución de tal experimento puede verse en la Figura 18.

Primero, hay que determinar la distancia fija a la que va a estar situada la cámara del sujeto a monitorizar. Aunque la cámara puede captar distancias entre 0,5 y 4,5 m. La página oficial de Microsoft señala que la distancia óptima para la Kinect® 2 es entre 0,8 y 1,8 m. Para un amplio movimiento del brazo sin salirnos del campo de visión, se escoge 1,7 m como distancia entre la cámara y el sujeto.

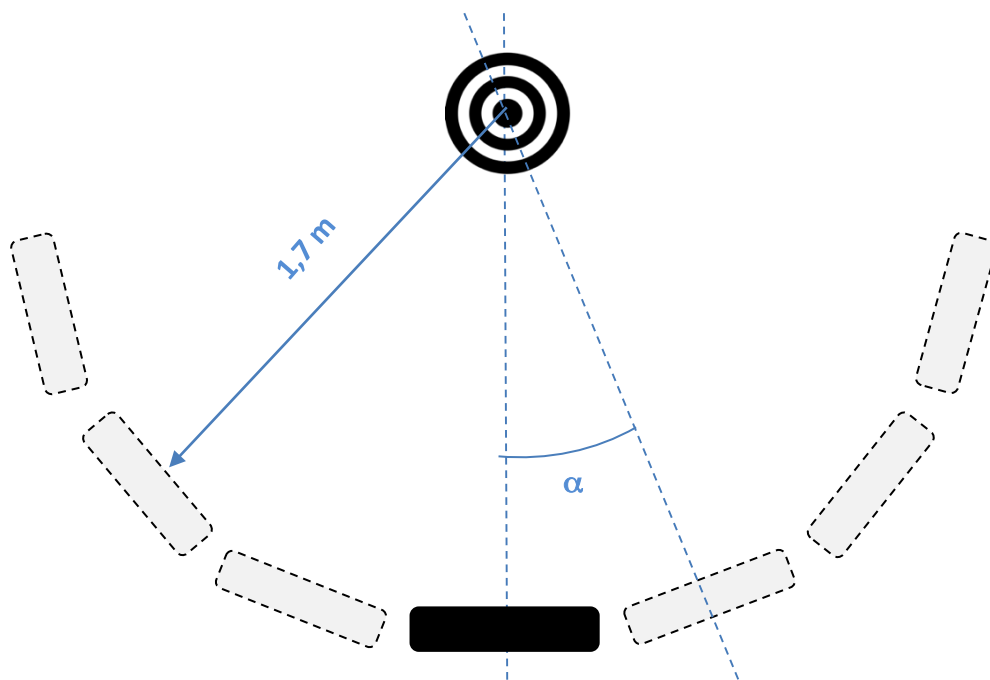


Figura 18. Distribución del espacio de los experimentos de configuración



Con el fin de ilustrar la influencia de la distancia, se muestran a continuación las consecuencias de la monitorización a distancias fuera de rango, donde vemos que no se reconoce el cuerpo en su plenitud o parcialmente (Figura 19 y 20).

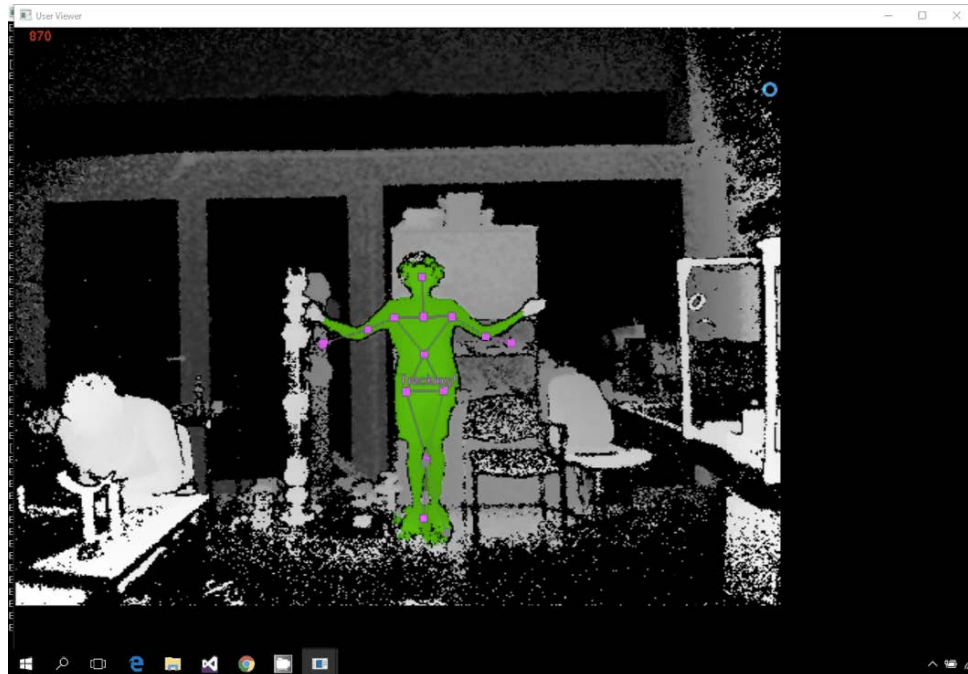


Figura 19. Monitorización a gran distancia

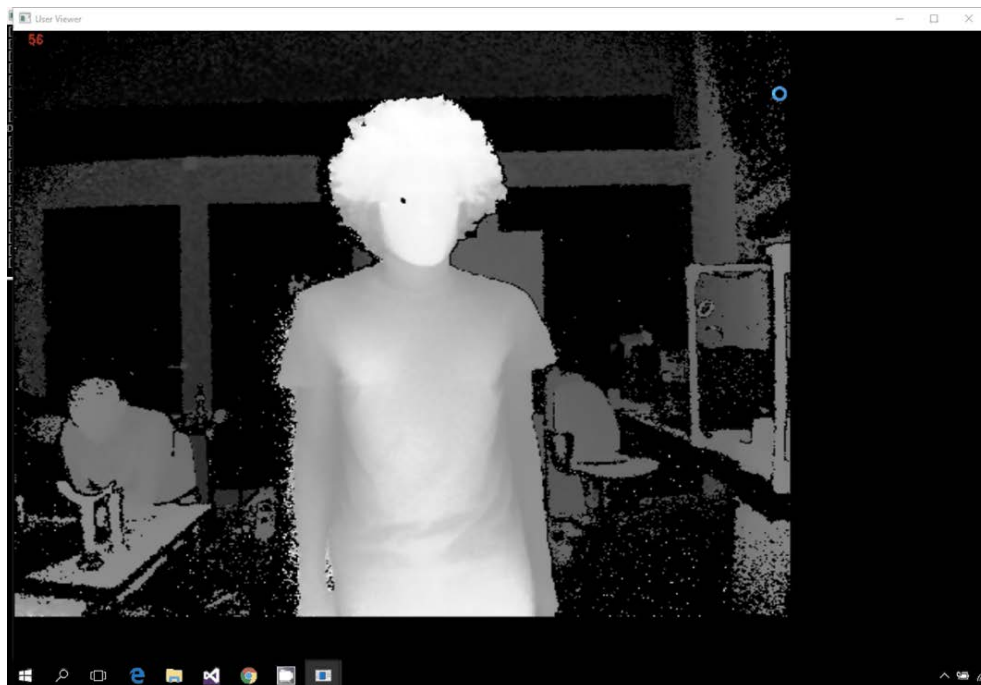
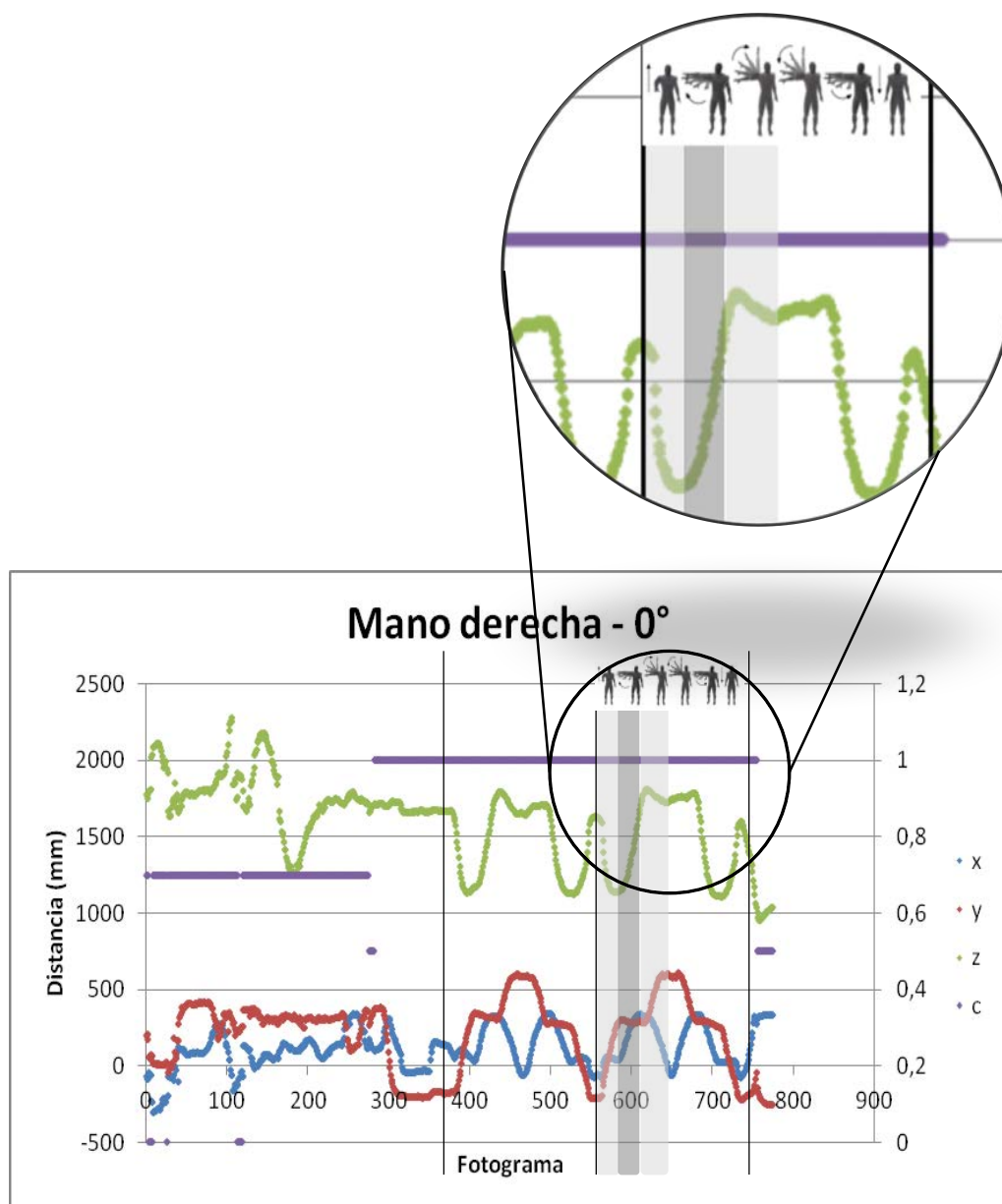


Figura 20. Monitorización a muy corta distancia

Fijada la distancia, ya se puede ir variando el ángulo  $\alpha$  para ver cómo se comporta el algoritmo y buscar el ángulo óptimo  $\alpha$ .

Se muestran a continuación los gráficos de los experimentos en los que se busca ese ángulo. Se explica justo a continuación de las figuras 21, 22 y 23 el proceso experimental, y se exponen las principales observaciones.



**Figura 21.** Gráfico de seguimiento para ángulo 0° diferenciando las distintas etapas de la secuencia de movimientos

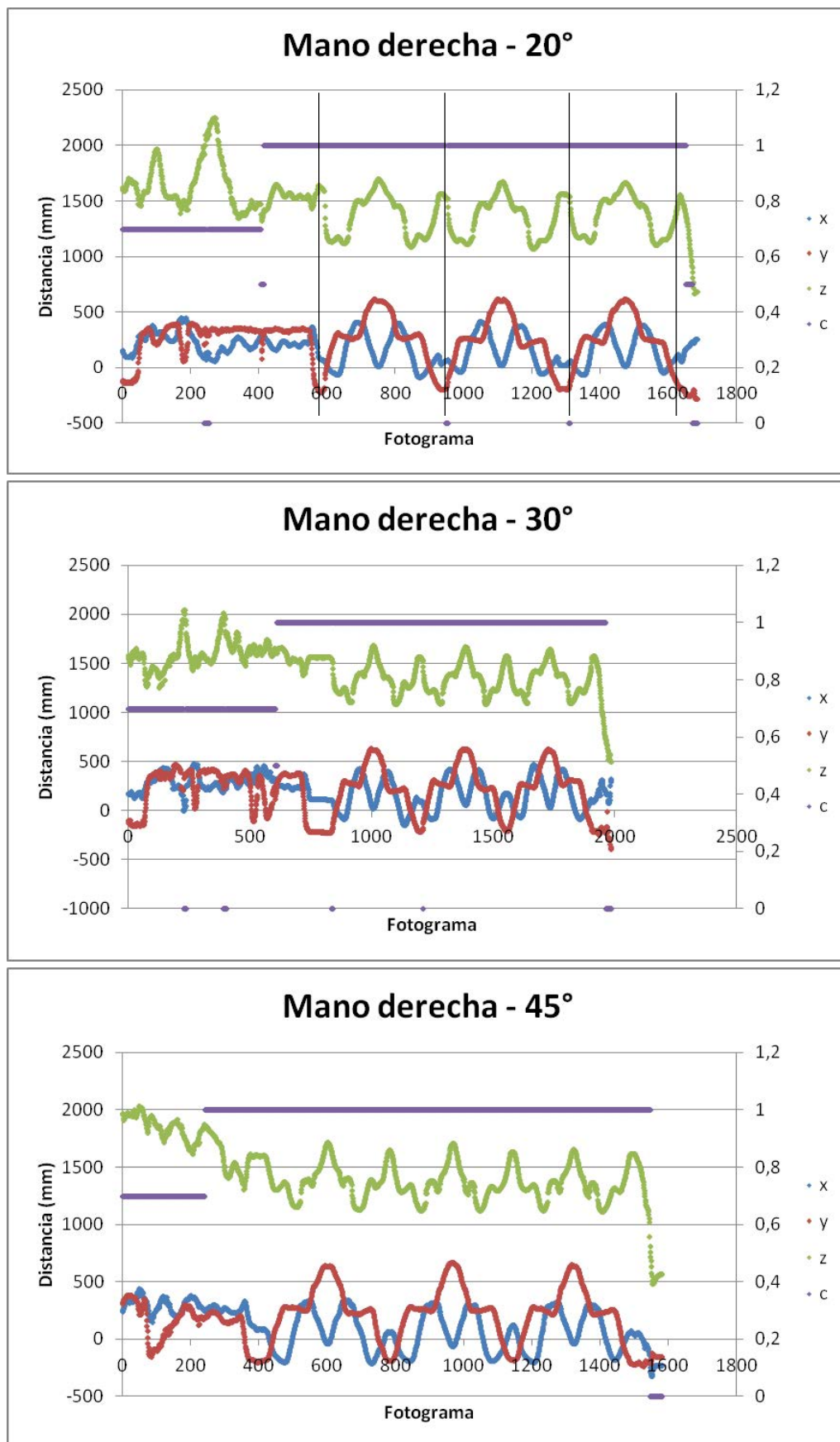


Figura 22a. Gráficos de seguimiento para distintos ángulos

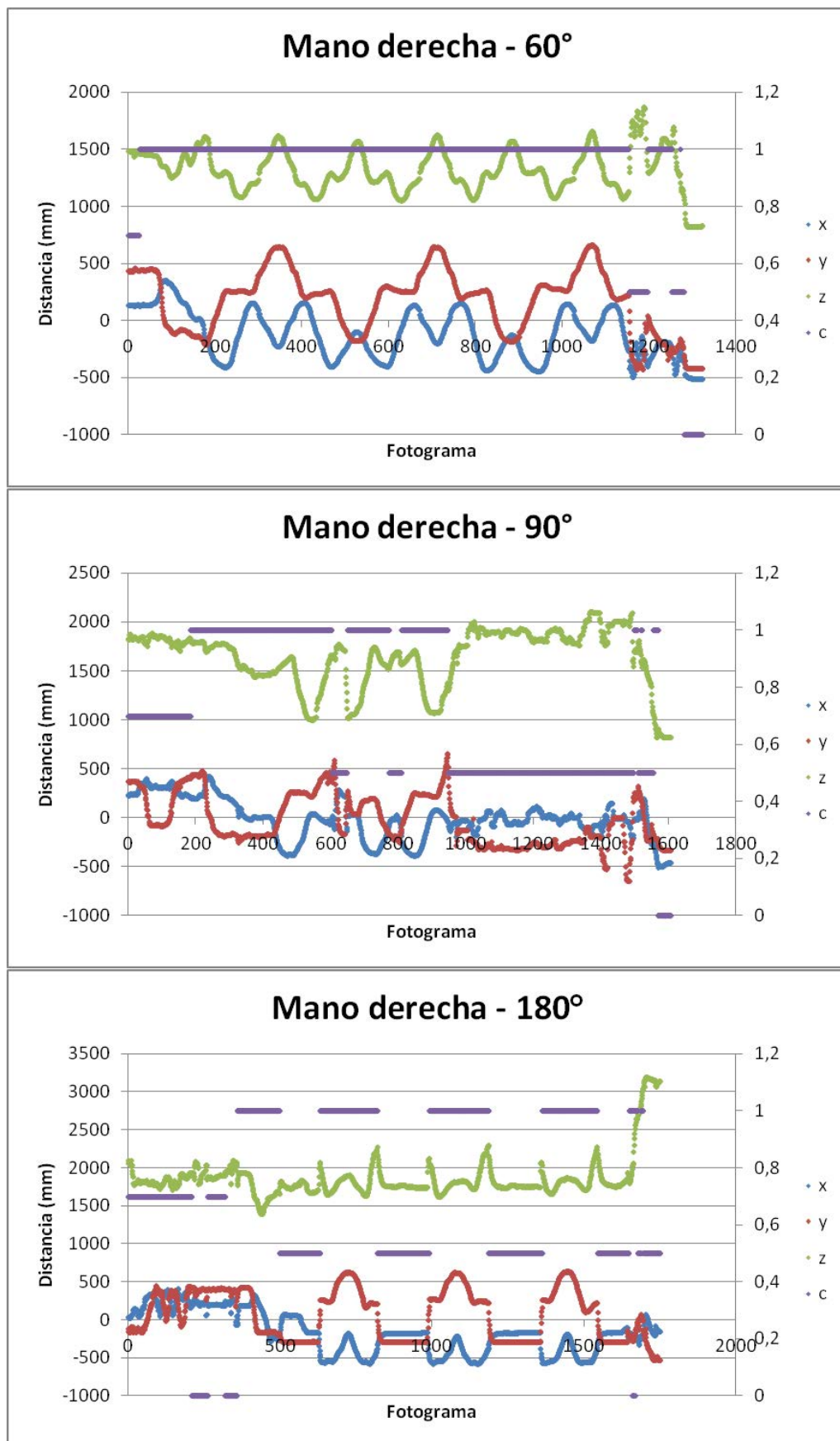


Figura 22b. Gráficos de seguimiento para distintos ángulos

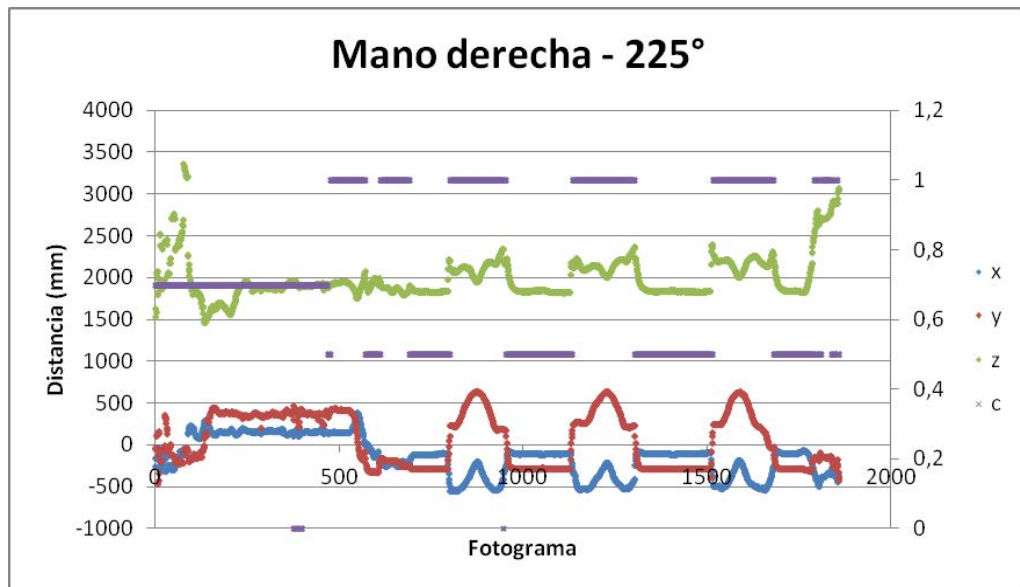
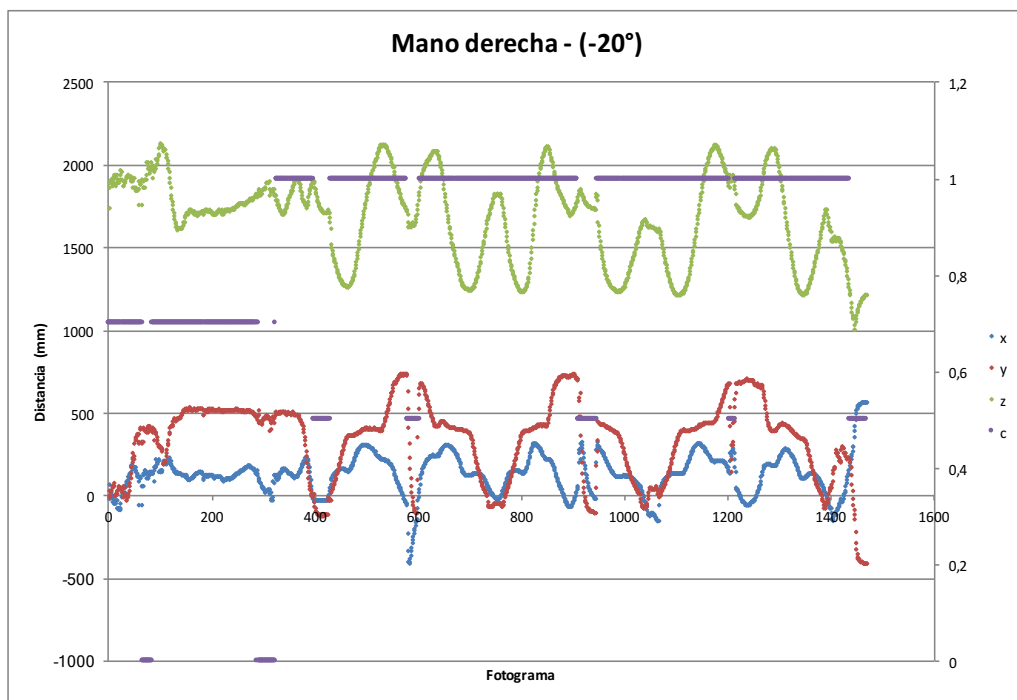
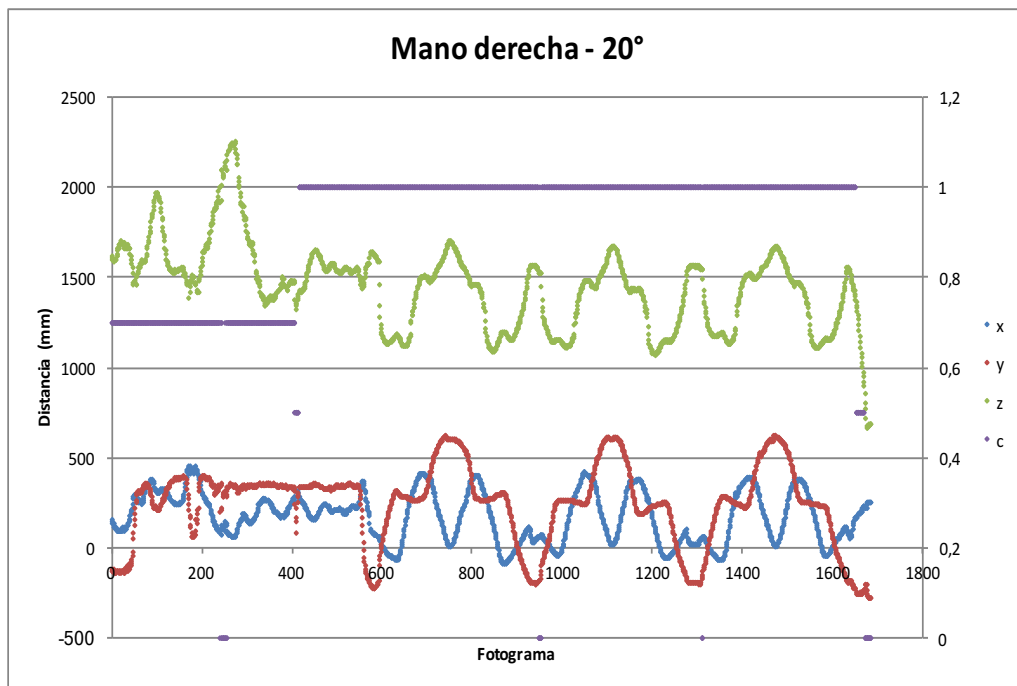
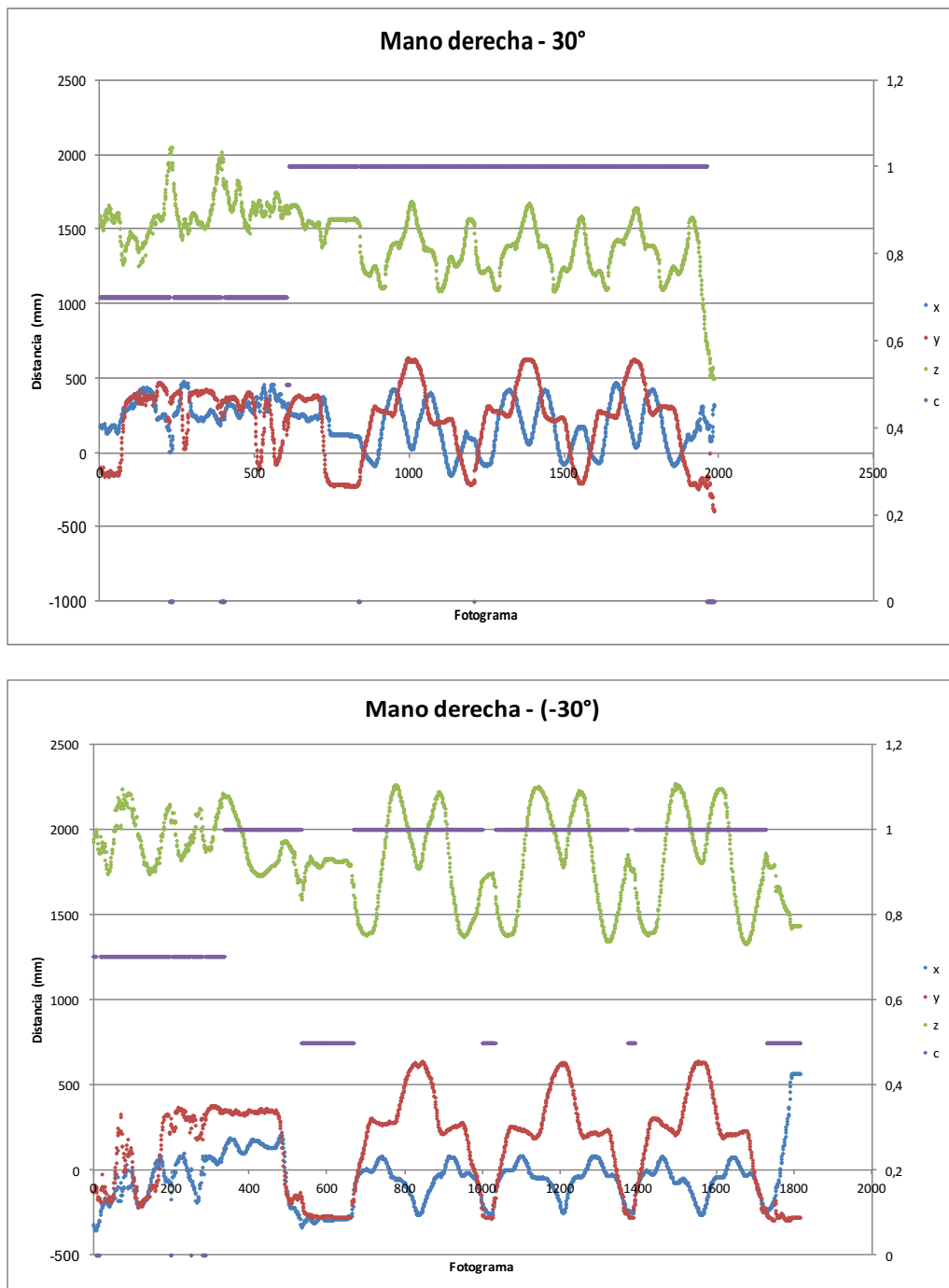


Figura 22c. Gráficos de seguimiento para distintos ángulos

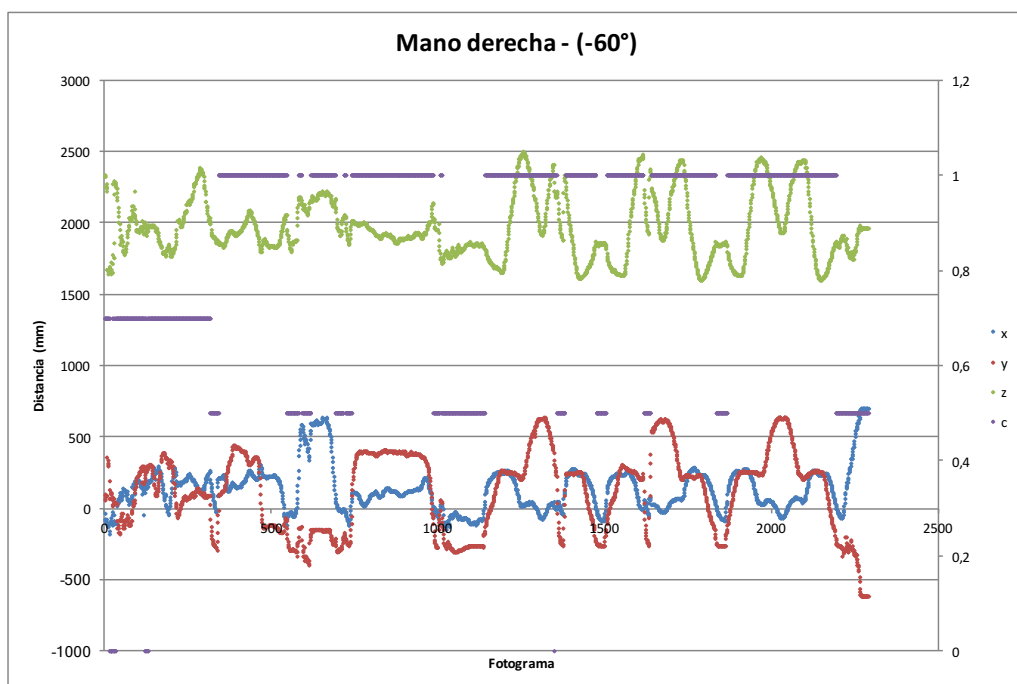
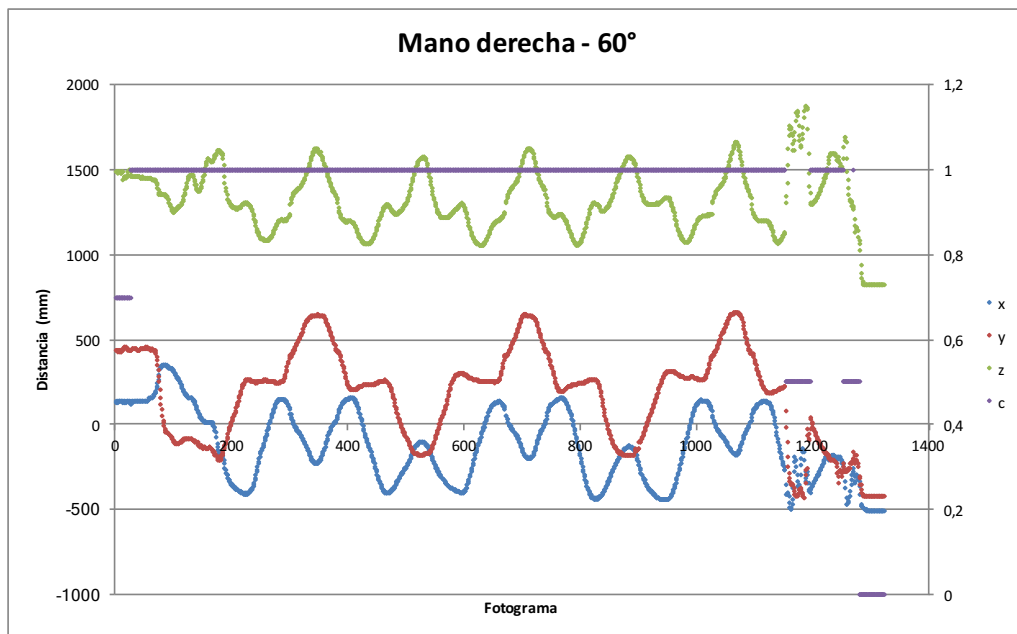


**Figura 23a.** Gráficos de seguimiento para las parejas simétricas de 20, 30 y 60 grados



**Figura 23b.** Gráficos de seguimiento para las parejas simétricas de 20, 30 y 60 grados





**Figura 23c.** Gráficos de seguimiento para las parejas simétricas de 20, 30 y 60 grados

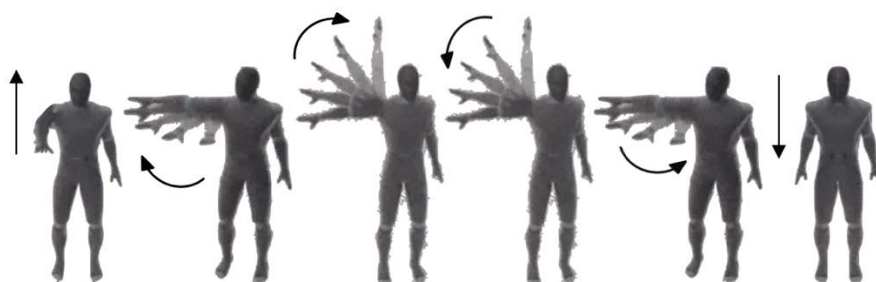


En todas las figuras anteriores (Figura 21,22 y 23) se muestra la posición en x, y, z y la confianza (c) de NiTE® para la articulación “Mano derecha”. Primero con una única cámara variando el ángulo entre 8 valores distintos. Después con dos cámaras en las parejas de ángulo de 20, 30 y 60 grados.

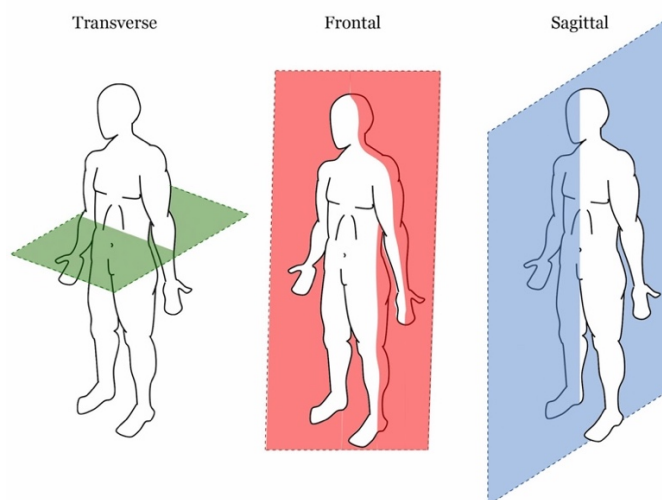
Se realizan tres repeticiones (excepto para cero grados que solo son dos repeticiones) de la secuencia contenida entre las barras negras. Dichas barras sólo se muestran en los dos primeros gráficos, pues pueden ser extrapolables a los demás de forma sencilla por la repetición de la forma de la curva.

Esta secuencia (ver Figura 24) consta de una subida del brazo en el plano sagital hasta el pecho, un giro en el plano transversal hasta quedar en forma de semi-T y una nueva subida hasta apuntar al techo, esta vez en el plano frontal (ver Figura 25 sobre planos corporales humanos). La secuencia termina repitiendo esos tres movimientos a la inversa.

La secuencia y los gráficos se corresponden tal y como muestra la Figura 21 donde la superposición de la Figura 24 ayuda a su entendimiento.



**Figura 24.** Secuencia de movimientos del experimento

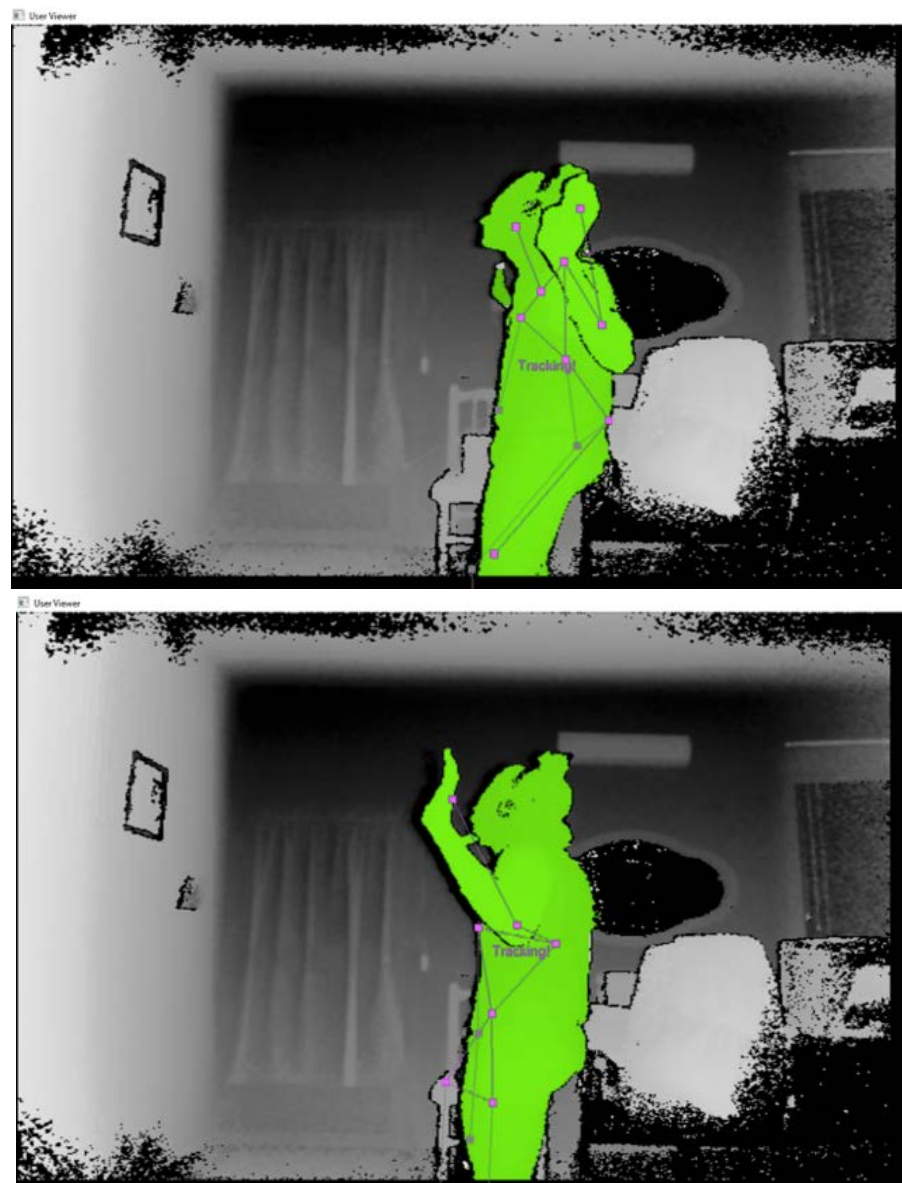


**Figura 25.** Planos corporales humanos

De los 14 experimentos se infiere que:

- **Figura 21:** A partir de 90 grados se pierde muchas veces al esqueleto, y aunque podrían usarse cámaras en estos ángulos como apoyo puntual a otras más frontales, se ve visualmente en el programa de captura con visualizador ("UserViewer") que el algoritmo no dibuja un buen esqueleto en estos ángulos, por lo que podrían obtenerse información falsa.

El algoritmo está diseñado para trabajar frontalmente y con pequeños giros, por lo que cuando el sujeto gira se observan esqueletos erróneos e inestables. Se descartan por tanto ángulos entre 90 y 270 grados (ver Figura 26 y 27).



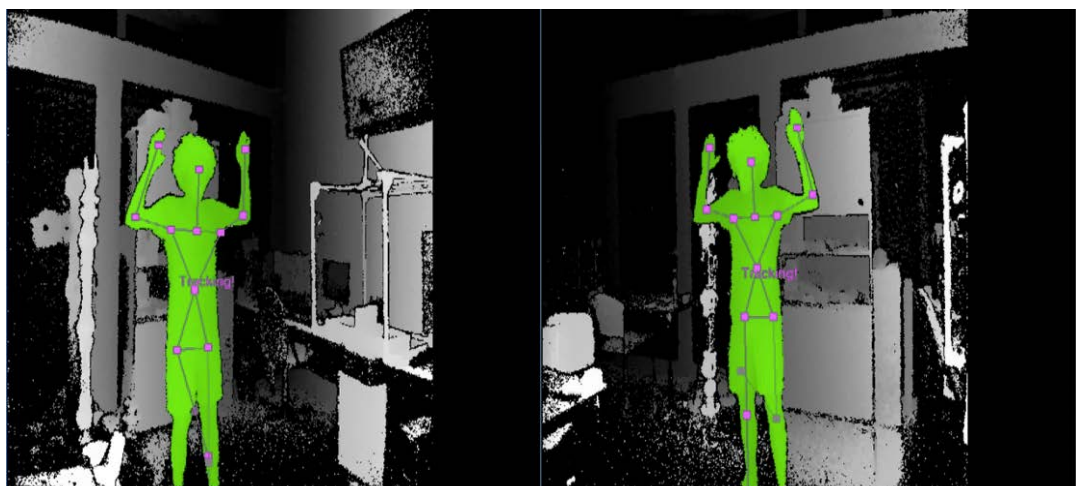
**Figura 26.** Intento de adaptación de esqueleto lateral (90 grados)



**Figura 27.** Intento de adaptación de esqueleto trasero (180 grados)

- **Figura 22:** Se seleccionan un ángulo bajo, uno intermedio y uno alto entre 0 y 90. Observamos una correcta repetibilidad y seguimiento de la mano para todos los ángulos.

Es solo visualmente que se puede notar una mejor adaptación del esqueleto en los ángulos de 20 y 30 grados debido a, como ya se ha discutido anteriormente, ser posiciones más frontales (ver Figura 28).



**Figura 28.** Esqueleto correctamente adaptado en dos cámaras a 20 grados cada una

### 3.4. Procesamiento e interfaz

Bien, ahora que se conoce el comportamiento de las cámaras con el algoritmo de NiTE® ante distintas configuraciones de ángulos, hay que calibrar el sistema, fusionar la información de ambas cámaras, mostrarla de forma clara y poner los filtros y condiciones necesarios para optimizar los resultados.

#### 3.4.1. CALIBRACIÓN DE LAS COORDENADAS

Antes de todo, como cada cámara tiene su sistema de referencia, los datos que se obtienen de cada una no pueden ser puestos en común. Antes habrá que fijar un sistema de referencia global sobre el que trabajar.

Para ello, se ha decidido que para nuestro caso las cámaras no se van a mover y el sistema de referencia del sistema estará en el punto medio entre las cámaras, para ver al objetivo de la monitorización (representado por la diana en la Figura 29) frontalmente. Entre este sistema global y el objetivo, se situará un brazo robótico cuando finalmente probemos este proyecto en un caso real de ejercicios de rehabilitación asistidos por robot.

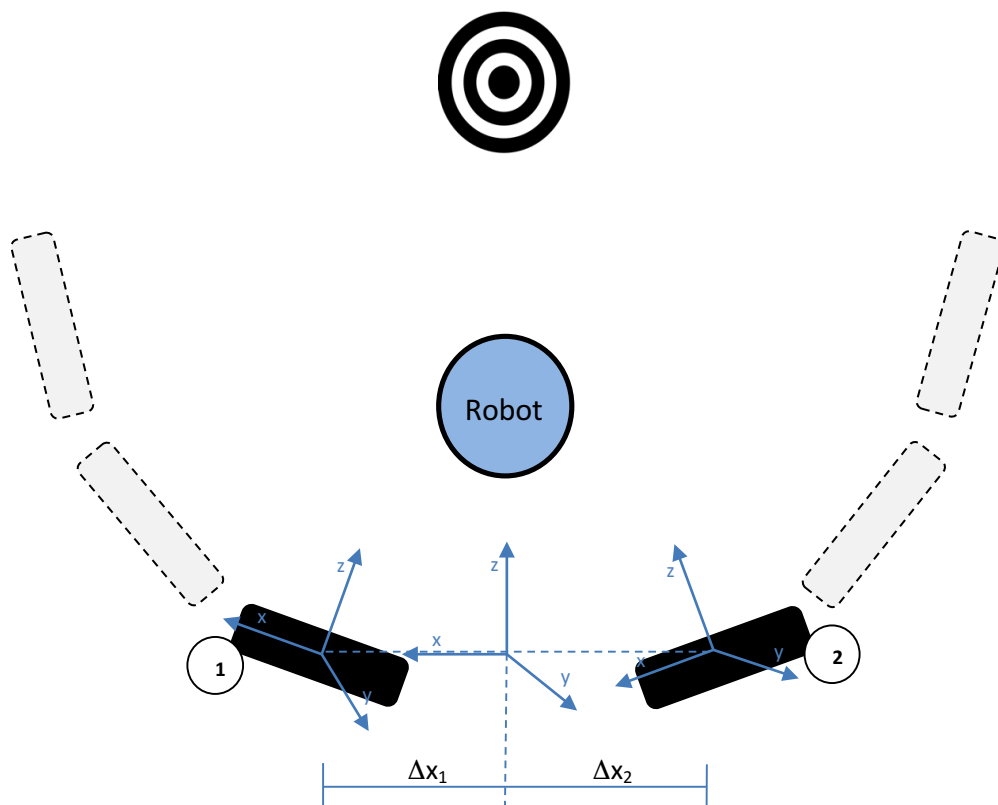


Figura 29. Mapa de coordenadas del sistema y configuración de las cámaras

Con el fin de cambiar todas las coordenadas a tal citado sistema global, se utilizará la matriz de transformación de la Figura 30.

$$\mathbf{T}(\mathbf{y}, \phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi & \Delta x \\ 0 & 1 & 0 & \Delta y \\ -\sin \phi & 0 & \cos \phi & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Figura 30.** Matriz de transformación

La presente matriz es la propicia para representar el desplazamiento de los sistemas en xyz y una rotación respecto al eje y. Sin embargo, debido a que las cámaras están situadas a la misma altura y profundidad, y a la simetría angular, sólo tendremos que deshacer el giro en el eje y, y calcular la distancia entre cámaras donde  $\Delta x_1 = \Delta x_2$ .

Este proceso aunque en un principio era manual, se ha automatizado mediante la función **Calibration**, donde se obtienen los parámetros de la matriz de transformación para después transformar las coordenadas con la función **Transform2Global**.

La función **Calibration** es llamada desde el botón amarillo de la interfaz con el mismo nombre, por lo que pueden recalcularse los parámetros de la matriz de nuevo durante la ejecución del programa.

Para obtener dichos parámetros es necesaria la función **antiTransform2Global** para deshacer el giro y obtener los desplazamientos entre los dos estados de las cámaras en el momento de la calibración.

En cuanto a procesos de calibración, cabe recordar en este punto que el algoritmo de NiTE realiza un proceso de calibración automática para adaptar un esqueleto a un cuerpo.

Por lo tanto, al principio, hasta que las articulaciones tienen 100% de confianza, conviene situarse de frente a las cámaras sin obstáculos y a poder ser en posición PSI para una mejor adaptación del esqueleto. Una vez en rosa es cuando podemos aplicar la calibración que hemos creado y procesar los datos de las cámaras.

### 3.4.2. COMPOSICIÓN DEL NUEVO ESQUELETO

Una vez en el mismo sistema de coordenadas, se ha pensado que la mejor forma de presentar la información es la siguiente.

Se comparará articulación por articulación en los dos esqueletos para quedarnos con la que mayor confianza tenga, de esta forma se creará entre las 30 articulaciones que obtenemos de las dos cámaras un nuevo esqueleto de 15.

En caso de tener el mismo valor de confianza, la nueva articulación se computará como una media de las de las cámaras.

Para más información, consultar código en Github [25].

En esta composición del nuevo esqueleto, se han eliminado las articulaciones inferiores de las rodillas y los pies por ser muy sensibles e inestables y porque nos hemos centrado en ejercicios del tren superior.

### 3.4.3. CÁLCULO DE ÁNGULOS

Para el cálculo de los ángulos creamos la función ***AnglesCalc*** que basándose en las coordenadas xyz de dos puntos calculará los ángulos proyectados sobre los planos xy, yz, zx utilizando razones trigonométricas.

Así, calcularemos los movimientos relativos entre articulaciones: cabeza-cuello, codo-hombro, mano-codo y cuello-torso.

También tenemos la función ***angle2vectors*** para calcular el ángulo entre dos vectores. Es la utilizada para el cálculo de los ángulos en “*3Joints Angles*” de la interfaz que se verá a continuación. Es decir, ángulos entre tres articulaciones como el del codo respecto a mano y hombro o el del hombro entre codo y la vertical del cuerpo.

### 3.4.4. INTERFAZ

Una vez procesada toda la información, se piensa que la mejor forma de mostrarla es la creación de una interfaz en MATLAB (Figura 31 y 32).

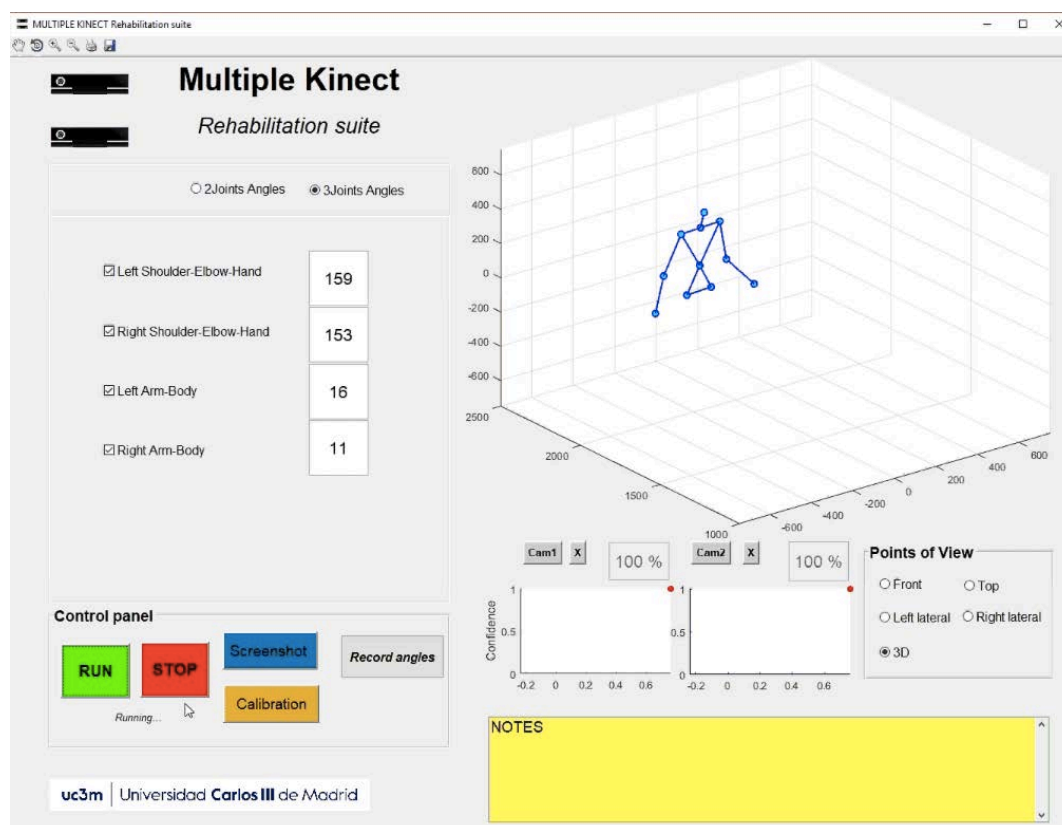


Figura 31. Interfaz gráfica de la utilidad de monitorización

| <input checked="" type="radio"/> 2 Joints Angles <input type="radio"/> 3 Joints Angles |       |         |     |
|--|-------|---------|-----|
|  | Front | Lateral | Top |
| <input checked="" type="checkbox"/> Head-Neck  | 84    | 86      | -35 |
| <input checked="" type="checkbox"/> Neck-Torso   | 88    | 89      | 37  |
| <input checked="" type="checkbox"/> Left elbow   | -74   | -84     | 20  |
| <input checked="" type="checkbox"/> Left hand  | -67   | -72     | -37 |
| <input checked="" type="checkbox"/> Right elbow  | -78   | -85     | 22  |
| <input checked="" type="checkbox"/> Right hand   | -64   | -77     | -25 |

Figura 32. Pestaña 2 Joints Angles no mostrada en Figura 30

Esta interfaz se compone de:

- **RUN:** botón que inicia el bucle while para lectura, procesado y visualización tan rápido como sea posible.
- **STOP:** botón que al pulsarlo rompe el bucle while. RUN y STOP pueden pulsarse tantas veces como se quiera para parar o iniciar el bucle.
- **Screenshot:** botón que guarda una captura de lo que se muestra en la pantalla en formato .png y además una copia de las variables en ese momento, de tal forma que puede ser cargada después para analizarla mejor.
- **Calibration:** botón que al pulsarlo durante ejecución recalcula los parámetros de la matriz de transformación para una correcta correlación de la información.
- **Record Angles:** botón que al pulsarlo durante ejecución comienza a guardar los ángulos seleccionados y el tiempo de ejecución en un archivo CSV. Para dejar de grabar un botón similar con el título "Stop recording" aparecerá y podrá ser pulsado para parar de guardar datos.
- **Cam1, Cam2 y sus botones "X":** el botón Cam1 inicia la captura de la cámara 1, Cam2 la captura de la cámara 2, y al pulsar "X" al lado de cada uno de los anteriores botones, se cierran los programas de captura abiertos, parando así la captura.
- **2Joints Angles:** esta sección de la izquierda te permite seleccionar los ángulos entre dos articulaciones que quieres medir para que se muestren sus valores. Se muestran las proyecciones sobre cada eje (x, y, z) del sistema global de referencia.
- **3Joints Angles:** esta otra sección te permite seleccionar los ángulos entre tres articulaciones para que se muestren sus valores, esta vez como ángulo en el espacio.
- **Gráfica principal:** (arriba derecha) se muestra el esqueleto recreado en tiempo real. Por defecto en 3D y con ejes prefijados.



- **Gráficas Confidence:** (encima de NOTES y a la izquierda de Points of View) muestran una confianza global de la cámara entre 0 y 1 para cada adquisición. Se calcula la media de las confianzas de las articulaciones visualizadas. También se muestra en valor numérico sobre cada gráfica.
- **Points of View:** se ofrece la posibilidad de cambiar el punto de vista de la gráfica principal para un mejor análisis del esqueleto (de momento sólo funciona parado, en tiempo real no se mantiene fija la vista seleccionada).
- **NOTES:** espacio creado para escribir comentarios durante la ejecución. Estas notas se guardan automáticamente en cualquier momento como archivo .txt en el directorio principal.
- **Barra de herramientas:** además, en la ventana se ha incluido una barra de herramientas que una vez parado el programa nos permite mover y rotar los gráficos, así como hacer zoom o imprimir.

Para más información consultar el código principal “interface.m” en GitHub [25].

### 3.4.5. INTEGRACIÓN

Llegados a este punto, se realizan varias pruebas del conjunto que integran los programas captors y la interfaz con su procesamiento para valorar su funcionamiento.

Se encuentran problemas con el rendimiento del ordenador. El MacBook Air no es capaz de correr con la carga de la adquisición, procesamiento y visualización de los captors, y además el procesamiento y visualización del programa en Matlab.

Lo máximo que se consigue hacer funcionar es una cámara y el programa en Matlab. Cuando se abre todo, uno de los programas acaba parándose.

Se ve obligado la búsqueda y uso de otro ordenador compatible con los drivers y más potente. Se hace entonces uso del ordenador del laboratorio de la marca MSI, donde todo corre perfectamente aunque se observa una alta carga para la CPU.

### 3.4.6. GUARDADO DE ÁNGULOS EN ARCHIVO CSV

Como uno de los objetivos era utilizar la monitorización del paciente para guardar datos sobre el ejercicio para un procesamiento posterior con el fin de evaluar la evolución, se necesita guardar los ángulos.

Para una fácil exportación e importación entre programas, se crea un archivo CSV. En él se guarda el tiempo de ejecución y todos los ángulos de la interfaz, los de la pestaña “3Joints Angles” que solo tienen una componente, se guardan tres veces en las tres filas de la estructura que se puede ver en la Figura 33.

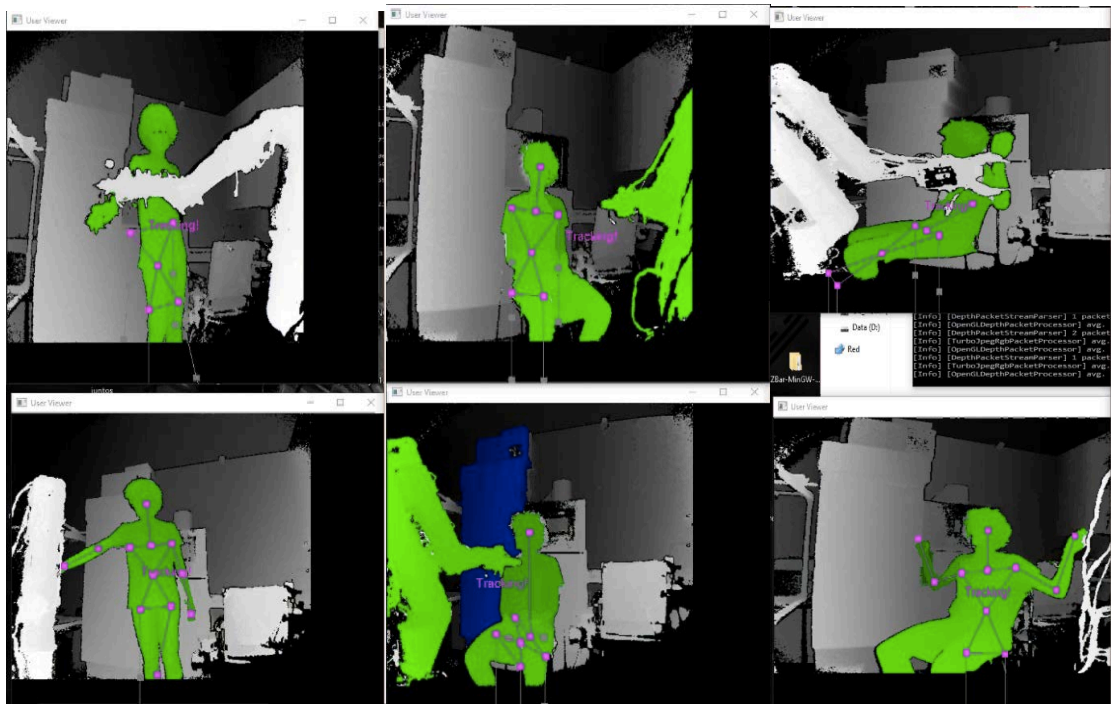
|    | A   | B | C | D | E | F | G | H | I | J | K |
|----|---|---|---|---|---|---|---|---|---|---|---|
| 1  | Time(s),head-neck,neck-torso,left elbow,left hand,right elbow,right hand,left shoulder-hand,right shoulder-hand,left arm-body,right arm-body(o) |   |   |   |   |   |   |   |   |   |   |
| 2  | Front,1.696,87,0,-89,0,0,-23,0,0,0,0  |   |   |   |   |   |   |   |   |   |   |
| 3  | Lateral,1.696,64,0,-90,0,0,-9,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 4  | Top,1.696,-84,0,-14,0,0,-70,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 5  | Front,2.001,87,0,-89,0,0,-23,0,0,0,0  |   |   |   |   |   |   |   |   |   |   |
| 6  | Lateral,2.001,64,0,-90,0,0,-9,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 7  | Top,2.001,-84,0,-14,0,0,-70,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 8  | Front,2.363,87,0,-89,0,0,-23,0,0,0,0  |   |   |   |   |   |   |   |   |   |   |
| 9  | Lateral,2.363,64,0,-90,0,0,-9,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 10 | Top,2.363,-84,0,-14,0,0,-70,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 11 | Front,2.648,87,0,-89,0,0,-23,0,0,0,0  |   |   |   |   |   |   |   |   |   |   |
| 12 | Lateral,2.648,64,0,-90,0,0,-9,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 13 | Top,2.648,-84,0,-14,0,0,-70,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 14 | Front,2.829,87,0,-89,0,0,-23,0,0,0,0  |   |   |   |   |   |   |   |   |   |   |
| 15 | Lateral,2.829,64,0,-90,0,0,-9,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |
| 16 | Top,2.829,-84,0,-14,0,0,-70,0,0,0,0   |   |   |   |   |   |   |   |   |   |   |

**Figura 33.** Ejemplo del fichero CSV donde se guardan los ángulos de la aplicación

### 3.4.7. FILTROS FINALES DE MEJORA EN LA CAPTURA

Durante los experimentos finales, cuando se lleva la solución al caso práctico, se detectan errores de detección debidos al movimiento y oclusión del brazo robótico durante los ejercicios. Estos errores no previstos inicialmente son corregidos aplicando una serie de condiciones y filtros a nuestra información captada.

El problema principal viene relacionado con la cabeza como puede verse en las siguientes tres parejas verticales de imágenes correspondientes a cada uno de los tres diferentes ejercicios (ver Figura 34).



**Figura 34.** Principales problemas con la cabeza del esqueleto de NiTE®

De izquierda a derecha, en la primera columna podemos observar que en el ejercicio de pie la cabeza no aparece en la vista de la cámara superior.

Por lo tanto, aunque las demás articulaciones están en rosa, es decir, el algoritmo cree estar seguro de su localización, hay que eliminar estas articulaciones erróneas. Con una sentencia “if” eliminaremos los datos de dicha cámara si se pierde la localización de la cabeza, es decir, si su confianza es cero.

La segunda pareja vertical muestra cómo el esqueleto se ha deformado y desplazado hacia abajo. La cabeza permanece en su sitio y el cuello por tanto se alarga de forma desmesurada. Se eliminarán entonces dichos puntos para la formación del esqueleto de la interfaz. Esto ocurrirá si la longitud del cuello es mayor que 250 mm (distancia razonable).

En la tercera columna el esqueleto está deformado y tumbado, resaltar que el segmento cuello-cabeza tiene un ángulo nunca logrado durante los tres ejercicios. Cabe señalar que es poco razonable para nuestro ámbito de aplicación que la cabeza forme más de 30 grados con respecto a la vertical y respecto al cuello.

Se descartarán por tanto aquellos esqueletos cuyo segmento cuello-cabeza forme menos de 60 grados con respecto a la horizontal en el plano XY y el plano YZ.

### 3.5. Diagramas de flujo

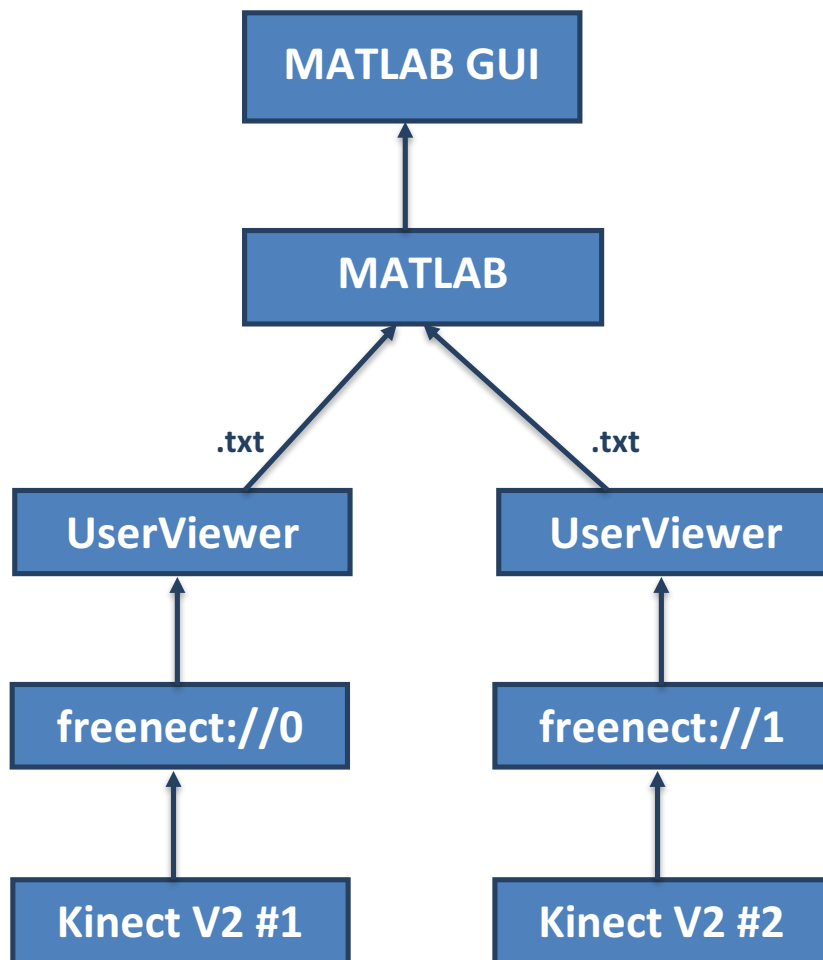


Figura 35. Mapa de la solución

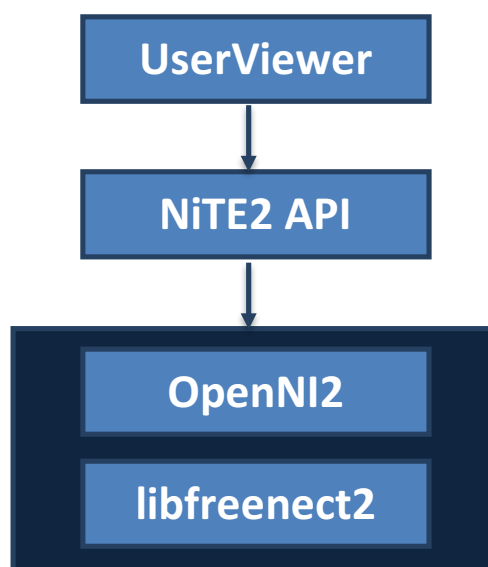


Figura 36. Mapa del programa UserViewer

### 3.6. Diagrama de Gantt

Para una vista de pájaro de la organización del proyecto se muestra aquí su diagrama de Gantt, sobre todo para ver en resumen las tareas involucradas en el proyecto y la duración relativa entre ellas.

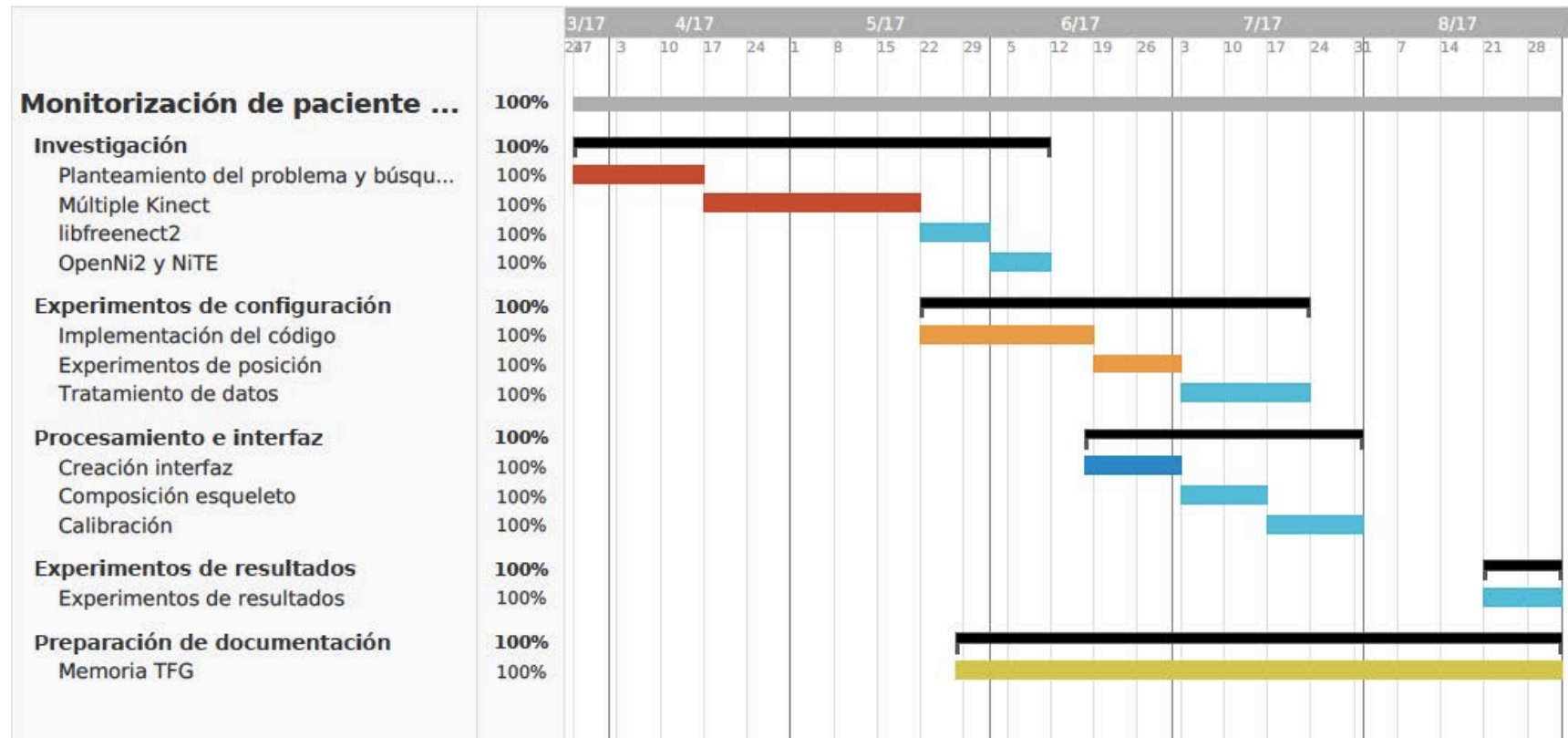


Figura 37. Diagrama de Gantt del proyecto (creado con la versión gratuita de teamgantt®)

# Capítulo 4

## Resultados

Se exponen en el presente capítulo los resultados obtenidos tras el largo proceso de desarrollo de la solución realizando experimentos de precisión angular, de comparación entre una y dos cámaras y por último, la realización de tres casos prácticos.

### 4.1. Ángulos realidad vs. Solución

En primer lugar, se centrará la atención en la precisión con la que se miden los ángulos. Para ello, se dibuja una semicircunferencia en la pared, se marcan los ángulos con reglas y transportador de ángulos, para ya ir situando el brazo izquierdo a diferentes posiciones angulares y anotar los ángulos medidos en ambos sistemas, pared y programa.

Comparamos así la diferencia entre el ángulo real y el medido con nuestra solución computacional (ver Figura 38).

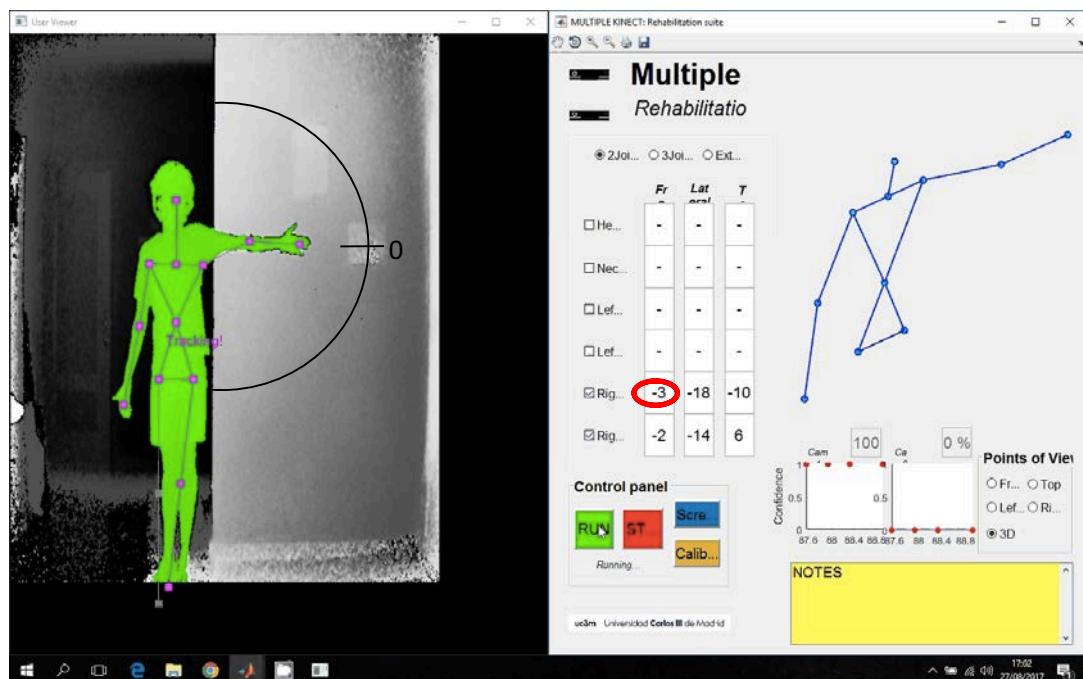


Figura 38. Experimento de precisión en la medida de los ángulos

Los datos experimentales son los siguientes (Tabla 2):

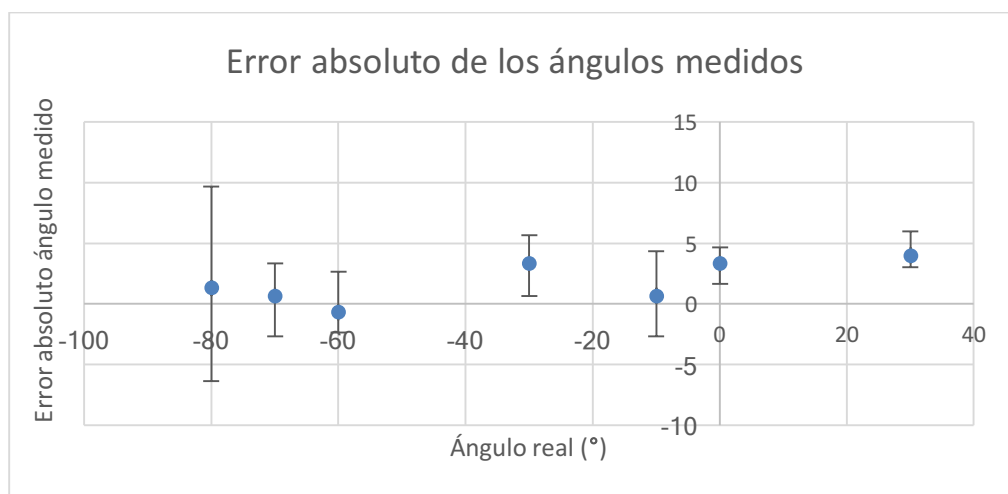
| Ángulo real (°) | Ángulo medido experimento 1 (°) | Ángulo medido experimento 2 (°) | Ángulo medido experimento 3 (°) |
|-----------------|---------------------------------|---------------------------------|---------------------------------|
| -80             | -82                             | -89                             | -73                             |
| -70             | -70                             | -68                             | -74                             |
| -60             | -61                             | -56                             | -61                             |
| -30             | -33                             | -36                             | -31                             |
| -10             | -11                             | -14                             | -7                              |
| 0               | -3                              | -2                              | -5                              |
| 30              | 25                              | 25                              | 28                              |

**Tabla 2.** Resultados experimentales en la medición de ángulos

Plasmamos los resultados medios y los errores en la siguiente tabla (Tabla 3).

| Ángulo real (°)  | Ángulo medio medido (°) | Error absoluto (°) |
|------------------|-------------------------|--------------------|
| -80              | -81                     | 1                  |
| -70              | -71                     | 1                  |
| -60              | -59                     | -1                 |
| -30              | -33                     | 3                  |
| -10              | -11                     | 1                  |
| 0                | -3                      | 3                  |
| 30               | 26                      | 4                  |
| Error máximo (°) |                         | 4                  |


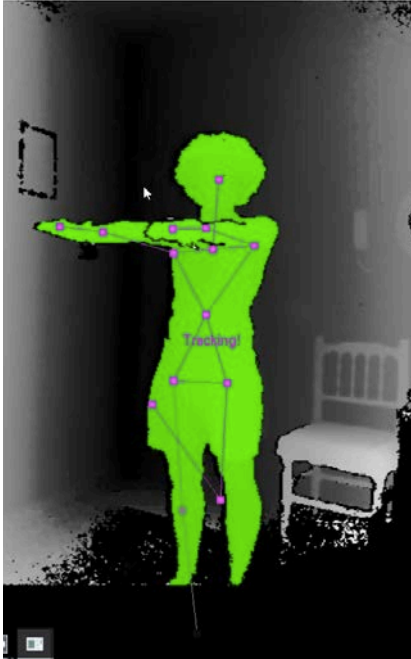

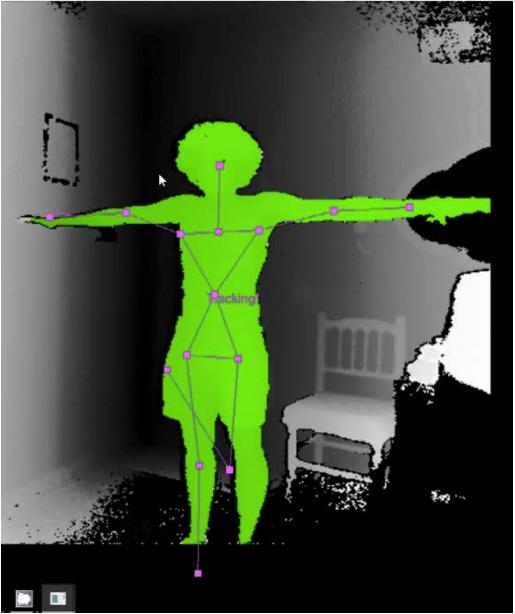
**Tabla 3.** Resultados medios en la medición de ángulos



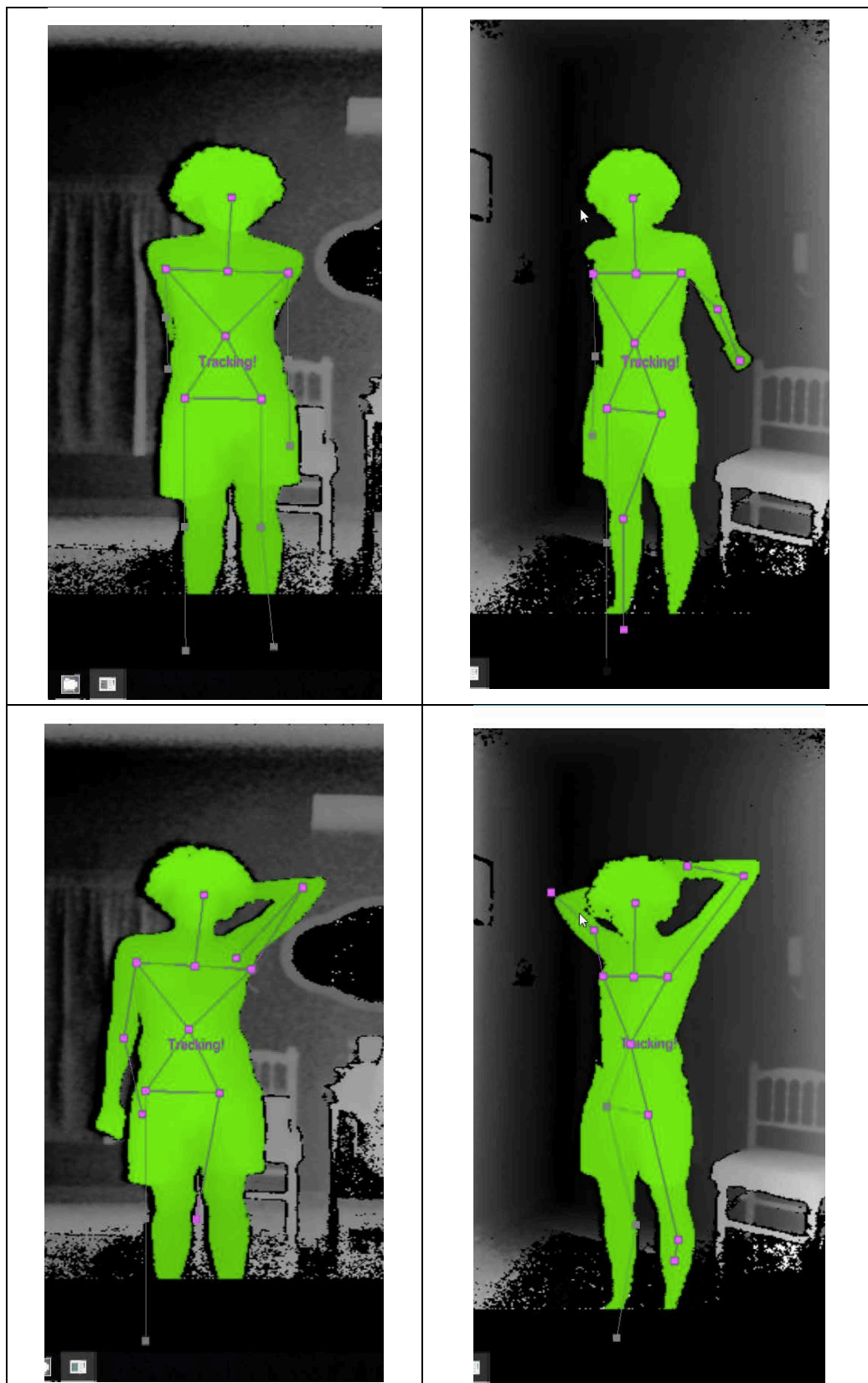
**Figura 39.** Gráfico comparativo entre ángulos reales y medidos con error

4.2. Múltiple cámara vs. Única cámara

Ahora, se compara la diferencia entre situaciones críticas con una y dos cámaras para ver las mejoras añadidas con la solución propuesta (ver Tabla 4).

| Única cámara  | Múltiple cámara  |
|---|--|
|   |   |
|  |  |





**Tabla 4.** Mosaico que muestra las mejoras con cámara múltiple versus única cámara

Aunque sólo se muestra una de las dos cámaras a la derecha, puede verse que al poner simplemente las cámaras a un cierto ángulo se evita el problema de la auto-oclusión en los movimientos transversales que ahora dejan de serlo.

También hay una mejora del ángulo de visión, ahora pueden monitorizarse los brazos cuando se sitúan levemente detrás del cuerpo. Desde cada uno de los dos lados se muestra mejor el brazo más cercano, mejorando ese ángulo de visión cuando se ponen en común.

#### **4.3. Caso práctico: ejercicios reales**

Dentro del marco de colaboración entre la Universidad Carlos III de Madrid con un hospital de Madrid, se tiene conocimiento del procedimiento de tres ejercicios de rehabilitación.

Estos ejercicios consisten en pedir al paciente alcanzar ciertos puntos en el espacio de menos a más alejados, en los cuales tendrá que coger una botella o enseñar algo a la cámara integrada en el brazo robótico del laboratorio, el cual ya tiene las posiciones de esos puntos guardadas.

Es interesante destacar la postura del paciente en los tres ejercicios. En el primero, de pie, en el segundo, sentado de frente, y el último, sentado y de lado; tal como puede verse en la Figura 34 del punto 3.4.7.

Las primeras pruebas reportaron varios fallos, de ahí la necesidad de añadir los filtros de mejora presentados igualmente en el punto 3.4.7.

Tras añadir tales mejoras, se pone a prueba de nuevo todo el sistema, resultando en una mejora notable como puede verse en la Figura 40 (girada), donde puede verse que aunque una de las cámaras detecta mal el esqueleto la otra lo hace perfectamente y el esqueleto resultado representado en azul en la interfaz responde mostrando el esqueleto sin errores.

Se observa también que los ángulos calculados tienen sentido. Los ángulos de  $176^\circ$  y  $167^\circ$  representan los ángulos que forma el codo con el hombro y la mano, izquierdo y derecho respectivamente. Se observa que los brazos están estirados, luego es coherente que el ángulo sea cercano a 180 grados.

El ángulo de 20 y 75 grados corresponden al que forma el codo con el tronco, izquierdo y derecho respectivamente. Puede comprobarse también que podrían perfectamente tratarse de esos valores exactos.

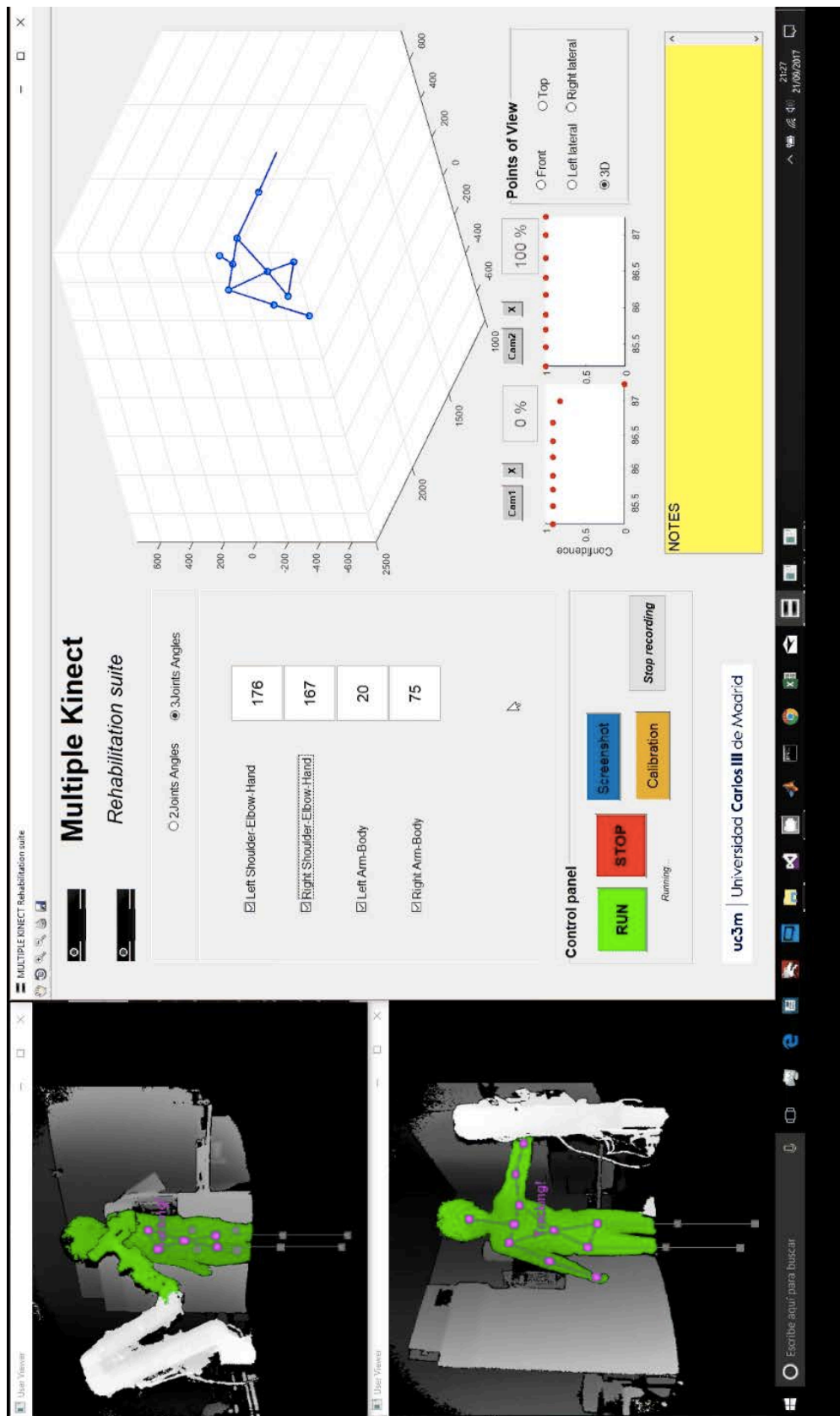


Figura 40. Prueba final de uno de los ejercicios reales

# Capítulo 5

## Marco regulador

Se tratan aquí todos los aspectos relacionados con legislación y normativa aplicable al proyecto, si se trabaja o se tienen que cumplir con estándares técnicos, y una aclaración de la propiedad intelectual.

### 5.1. Legislación aplicable

En el ámbito legal, el proyecto podría verse afectado principalmente por las leyes de Protección de Datos.

De acuerdo con la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal, podría carecer de legalidad la adquisición y grabación de datos del paciente.

Por lo tanto, se sugiere que ante posibles conflictos legales se requiera el consentimiento del paciente para la grabación y procesamiento de vídeo, así como datos derivados de estos como notas de evolución, datos clínicos u otros datos personales.

No existen otras responsabilidades profesionales o éticas, así como riesgos laborales, importantes.

### 5.2. Estándares técnicos

No se señalan estándares técnicos especialmente aplicados, mas cabe destacar que Microsoft Visual Studio recoge estándares del lenguaje C++. Concretamente, implementa la mayoría de las características de la especificación principal del lenguaje C++11, así como muchas características de la biblioteca de C++14 y algunas de las características propuestas para C++17.

### 5.3. Propiedad intelectual

En primer lugar, se reconoce al autor de este proyecto, yo, Jorge Simal, la autoría del código de procesamiento y ciertas modificaciones en los códigos del programa captor.

Dicho programa ha sido creado a raíz de una modificación de un ejemplo distribuido por el middleware NiTE2, cuya licencia se describirá a continuación. Cabe destacar la adquisición reciente de PrimeSense® por parte de Apple Inc. lo que deja un panorama confuso en cuanto a términos de licencia del software adquirido con anterioridad.

Se seguirán por tanto los términos y condiciones que venían con el software instalado, los cuales nos conceden una licencia limitada, mundial, no exclusiva, personal, no transferible y revocable para el uso del middleware NiTE® [21].

Asimismo, el programa hace uso de las librerías de código abierto libfreenect2 y OpenNI2 que se suscriben bajo licencia apache para uso perpetuo, internacional, no exclusivo, sin cargo e irrevocable [22][23]. Mucho más permisiva que la anterior para NiTE®.

Por otro lado, para el uso de MATLAB® hemos utilizado una licencia de estudiante que la Universidad Carlos III de Madrid compra para sus alumnos. Esta licencia es bastante restrictiva en cuanto, por ejemplo, hacer uso comercial del programa. Por lo que si quisiéramos venderla o algo así, tendríamos que comprar otra licencia.

# Capítulo 6

## Entorno socio-económico

En este capítulo, se expone un análisis del impacto que el proyecto puede causar a nivel social y económico. De esta forma, a la hora de implementarlo, se conocerán los riesgos medioambientales, sociales, éticos y económicos.

Este análisis es importante en la medida en la que el desarrollo sostenible y el contexto en general se vuelven cada vez más importantes. Tal es ésta, que la mayoría de las empresas, hoy en día, tienen reflejados en sus páginas web sus compromisos y valores para con la sociedad y el medio ambiente, tratando no sólo de vender productos si no de aportar auténtico valor a la sociedad.

### 6.1. Impacto socio-económico

Tras los resultados obtenidos en el proyecto, creemos que con cierta optimización de los algoritmos de procesamiento, tanto este proyecto como otros similares pueden suponer una auténtica revolución en la monitorización de pacientes.

Económicamente, presenta un medio más asequible para usuarios y organizaciones con menores ingresos. Tanto para rehabilitación como para otras aplicaciones, como discutíamos en “Estado del arte”, tales como animación, escáner 3D, videojuegos, etc.

Socialmente, todas las tareas implicadas en mejorar procesos médicos conllevan una mejora social a nivel global, pues se llega a más personas y de forma más asequible.

En cuanto a impacto medioambiental, ha sido una grata sorpresa toparse con que existe una página dedicada a soporte de Xbox donde nos dan indicaciones para reciclar accesorios como la Kinect® [24]. Es imprescindible el desarrollo sostenible para un futuro mejor, y para ello tenemos que poner medidas y diseñar pensando en el fin de la vida del producto.

En términos de impacto ético, se suscita la tradicional polémica máquina-hombre relacionada con la automatización de trabajos que puede conllevar la pérdida de puestos de trabajo.

Y aunque esta herramienta y proyectos relacionados se proponen como un complemento, podría dar lugar a aplicaciones de uso a distancia o disminución de tareas auxiliares de medición, por lo que se contrataría menos personal auxiliar.

## 6.2. Presupuesto del proyecto

Pasamos a analizar los costes generados para la realización del presente proyecto. Para ello, en primer lugar, se valorarán los costes dependientes de la fase de desarrollo. Después, los costes de los recursos necesarios para su aplicación. El cómputo global generará los costes del proyecto.

Para alcanzar los objetivos planteados inicialmente siguiendo el desarrollo detallado en el Capítulo 3, se han empleado los siguientes tiempos para cada fase (ver Tabla 5).

| Fase   | Tiempo (h) |
|--|------------|
| Estudio teórico del entorno Kinect                           | 20         |
| Comprensión del trabajo a realizar y búsqueda de la solución | 110        |
| Implementación del código                                    | 90         |
| Pruebas de algoritmo   | 10         |
| Desarrollo de interfaz y procesamiento                       | 50         |
| Preparación del entregable y últimas pruebas                 | 20         |
| Redacción de documentación                                   | 100        |
| <b>TOTAL</b>   | <b>400</b> |

**Tabla 5.** Resumen del tiempo de trabajo del ingeniero

Para ello el ingeniero ha utilizado un ordenador propio cuyos gastos de amortización se imputan al proyecto, así como dos cámaras Kinect con su respectivo adaptador.

Tanto los costes de recursos como de mano de obra se computan a continuación (Tabla 6). Impuestos incluidos.

| Recurso  | Cantidad (u) | Precio   | Tiempo       | Coste (€)      |
|--|--------------|----------|--------------|----------------|
| Apple MacBook Air 13,3" MMGF2Y/A Intel Core i5, 128 GB SSD | -            | 220€/año | 4 meses      | 74,00          |
| Ingeniero  | -            | 20€/h    | 400 h        | 8000,00        |
| Cámara Kinect V2   | 2            | 99,99    | -            | 199,98         |
| Adaptador Kinect para PC                                   | 2            | 39,99    | -            | 79,98          |
| Soporte vertical Kinect                                    | 2            | -        | -            | 27,85          |
|  |              |          | <b>TOTAL</b> | <b>8381,81</b> |

**Tabla 6.** Resumen costes de desarrollo

Luego, el desarrollo del proyecto ha supuesto unos costes totales de 8381,81 €.



# Capítulo 7

## Conclusiones

Tras el desarrollo de la solución más apropiada, se aúnan en este capítulo las contribuciones conseguidas, las principales conclusiones y algunas sugerencias para desarrollos futuros en la línea de este proyecto.

### 7.1. Discusión final

Tras una larga investigación y varios esfuerzos por lograr la mejor solución a nuestro objetivo principal de monitorización de un paciente en rehabilitación asistida por robot, con una aceptable precisión y evitando oclusiones y ángulos muertos, se ha conseguido una aplicación que consigue tal cometido con un relativo bajo coste como muestra el presupuesto en 6.2.

Los objetivos secundarios del cálculo de ángulos del tren superior y la aplicación de métodos para la mejora de la precisión también se han llevado a cabo, aunque se podría seguir trabajando en este campo de la precisión para refinar todo aún más.

Hay que tener en cuenta ciertas limitaciones de los métodos y tecnologías escogidos como que a diferencia de las Kinect® de primera generación, las de segunda generación, objeto de nuestro estudio, presentan una mayor resolución y con ello un mayor y alto consumo de recursos computacionales.

Además, los fundamentos científicos del algoritmo de detección de esqueleto obligan a que las detecciones funcionen mejor cuanto más frontal es la vista del cuerpo humano. Por tal motivo se escogen ángulos inferiores a 30 grados para la distribución de las cámaras.

De los distintos experimentos, se observa una buena repetibilidad en las gráficas de los experimentos de configuración y fiabilidad que indican que habría poca variabilidad entre mediciones de movimientos iguales en el rango de ángulos de 20 a 30 grados principalmente.

En los experimentos con los que se muestran los resultados se ratifica que los objetivos se han cumplido, que efectivamente se elimina la oclusión, que hay

consistencia entre los ángulos calculados y los reales y que en un caso práctico la solución podría ser de utilidad. La precisión de la captura es notablemente buena con tan solo un error absoluto máximo de 4 grados.

La fiabilidad de estos resultados aunque parece aceptable debería ser contrastada con un sistema más preciso para medir su eficacia. En esta línea, el poder realizar un ensayo paralelo de seguimiento de movimientos con Kinect® (sistema sin marcadores) frente a Vicon® (sistema con marcadores) permitiría validar completamente la propuesta.

## **7.2. Desarrollos futuros**

Para proyectos futuros en esta línea de investigación, en primer lugar, se podría intentar mejorar aún más la presente solución, su robustez y precisión.

Por otro lado, y quizás sea mejor para comparar, se podría ahondar en la posibilidad de usar las herramientas oficial de Microsoft®. Aunque esta idea sea algo más cara y tediosa de llevar a cabo por su necesidad de ordenadores independientes, podría suponer mejoras importantes de estabilidad y detalle, pues recordemos que la SDK de Microsoft® permite calcular hasta 10 articulaciones más que las 15 que ya detecta Kinect®.

Por último, sería especialmente interesante que una vez se tuvieran herramientas bastante optimizadas con NiTE® y/o el SDK de Microsoft, se compararan con un sistema Vicon®. Al ser esta la solución referente en el sector daría una relación de destreza-coste que permitiría a nuestra solución ubicarla en el mercado. Y comparando también, localizar los fallos y puntos más críticos.

Lejos de la rehabilitación y monitorización, relacionado con la tecnología e impresión en 3D, se podrían utilizar una o varias cámaras del tipo Kinect® como escáner.

De esta forma, se podrían escanear desde personas a piezas industriales de las que se guardaría y superpondría su información de profundidad para después tratar con programas de modelado 3D y de aquí imprimir todo tipo de derivados con modificaciones, juguetes, figuritas de la persona en cuestión, etc.

# Bibliografía

- [1] Organización Mundial de la Salud, “Atención médica y rehabilitación”. [En línea]. Disponible en: <http://www.who.int/disabilities/care/es/>.
- [2] Elena Milla, “El futuro de las pensiones a causa del envejecimiento poblacional”, *Rankia*, 12-03-13. [En línea]. Disponible en: <https://www.rankia.com/foros/economia/temas/1711829-futuro-pensiones-causa-envejecimiento-poblacional>
- [3] A. Villarroja, A. Alonso, M.A. Franco, N. Ramírez-Escudero, I. García, B. Calvo, “Sensibilidad a la colocación de los marcadores en el modelo convencional de análisis cinemático de la marcha”, *Biomecánica*, 15 (1), pp. 42-48, 2007. [En línea]. Disponible en: <https://upcommons.upc.edu/bitstream/handle/2099/6800/09%20-%20Sensibilidad%20a%20la%20colocaci%C3%B3n%20de%20los%20marcadore s%20en%20el%20modelo%20convencional%20de%20an%C3%A1lisis%20cinem %C3%A1tico%20de%20la%20marcha.pdf>. Acceso: 26-09-2017.
- [4] Otte K, Kayser B, Mansow-Model S, Verrel J, Paul F, Brandt AU, et al. (2016), “Accuracy and Reliability of the Kinect Version 2 for Clinical Measurement of Motor Function”. *PLoS ONE* 11(11): e0166532. [En línea]. Disponible en: <https://doi.org/10.1371/journal.pone.0166532>
- [5] Brrok Galna, Gillian Barry, Dan Jackson, Dadirayi Mhiripiri, Patrick Olivier, Lynn Rochester, “Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson’s disease”, *Gait & Posture*, Volume 39, Issue 4, April 2014, pp. 1062-1068. [En línea]. Disponible en: <https://doi.org/10.1016/j.gaitpost.2014.01.008>. Acceso: 26-09-2017.
- [6] Microsoft Forum. “Multiple Kinect for Windows v2 Projects”. [En línea]. Disponible en: <https://social.msdn.microsoft.com/Forums/en-US/111cd7a0-0a42-456a-bc18-dd819a7d5b51/multiple-kinect-for-windows-v2-projects?forum=kinectv2sdk>.
- [7] Amine Magdich. “Programming Kinect V2 For Windows TUTO1”. *Youtube*, 20-03-2015. [Video en línea]. Disponible en: <https://youtu.be/GPjS0SBtHwY>.
- [8] Kwok-Yun Yeung, Tsz-Ho Kwok, and Charlie C.L. Wang, “Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications”, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. [En línea]. Disponible en: <http://homepage.tudelft.nl/h05k3/pubs/ICISERealTimeSkeleton.pdf>.
- [9] Sungphill Moon, Youngbin Park, Dong Wook Ko, Il Hong Suh. “Multiple Kinect Sensor Fusion for Human Skeleton Tracking Using Kalman Filtering”, Hanyang University, Seoul, Republic of Korea, 2016. [En línea]. Disponible en: <http://journals.sagepub.com/doi/pdf/10.5772/62415>

- [10] “Filtro de Kalman”, *Wikipedia*. [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Filtro\\_de\\_Kalman](https://es.wikipedia.org/wiki/Filtro_de_Kalman). Acceso: 26-09-2017.
- [11] Suttipong Kaenchan, Pornchai Mongkolnam, Bunthit Watanapa, and Sasipa Sathienpong. Automatic multiple Kinect cameras setting for simple walking posture analysis. <http://ieeexplore.ieee.org/document/6694787>
- [12] “Kinect”, *Wikipedia*. [En línea]. Disponible en: <https://en.wikipedia.org/wiki/Kinect> . Acceso: 26-09-2017.
- [13] Página web OpenKinect. [En línea]. Disponible en: [https://openkinect.org/wiki/Main\\_Page](https://openkinect.org/wiki/Main_Page) .
- [14] Lingzhu Xiang, , Florian Echtler, , Christian Kerl, , Thiemo Wiedemeyer, , Lars, , hanyazou, , ... Alistair, . (2016, April 28). libfreenect2: Release 0.2. Zenodo. <http://doi.org/10.5281/zenodo.50641>
- [15] Página de libfreenect2 en GitHub. [En línea]. Disponible en: <https://github.com/OpenKinect/libfreenect2>
- [16] OpenNI. Artículo de Wikipedia. <https://es.wikipedia.org/wiki/OpenNI>
- [17] Página de OpenNI en Structure. [En línea]. Disponible en: <https://structure.io/openni>
- [18] Sitio web que alberga una copia de NiTE. [En línea]. Disponible en: <http://openni.ru/files/nite/>
- [19] Sitio web oficial de Matlab. [En línea]. Disponible en: <https://es.mathworks.com/>
- [20] JSharp4273, “Multiple kinect v2 usb configuration issue”. [En línea]. Disponible en: <https://github.com/christiankerl/libfreenect2/issues/10>
- [21] Licencia NiTE. [En línea]. Disponible en: <http://openni.ru/nite-licensing-and-distribution-terms/index.html> .
- [22] Licencia libfreenect2. [En línea]. Disponible en: <https://github.com/OpenKinect/libfreenect2/blob/master/APACHE20> .
- [23] Licencia OpenNI. [En línea]. Disponible en: <https://github.com/OpenNI/OpenNI2/blob/master/LICENSE> .
- [24] Microsoft, “Recicla tu Xbox y sus accesorios”. [En línea]. Disponible en: <http://support.xbox.com/en-CA/my-account/warranty-and-service/recycle-your-xbox>
- [25] Repositorio Github del presente proyecto. [En línea]. Disponible en: <https://github.com/JorgeSimal/MKRS>